Robust Subspace Estimation via Low-Rank and Sparse Decomposition and Applications in Computer Vision

by

Salehe Erfanian Ebadi

Submitted in partial fulfillment of the requirements of the Degree of
Doctor of Philosophy

School of Electronic Engineering and Computer Science
Queen Mary University of London
United Kingdom

January 2018

To my parents,
my little brother, and my lovely sister-in-law.

Statement of Originality

I, Salehe Erfanian Ebadi, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without the prior written consent of the author.

Salehe Erfanian Ebadi

8th January 2018

Details of collaboration and publications:

All papers published while working on this thesis are listed at the end of the thesis. Any publication produced in collaboration with others is clearly mentioned.

Abstract

Recent advances in robust subspace estimation have made dimensionality reduction and noise and outlier suppression an area of interest for research, along with continuous improvements in computer vision applications. Due to the nature of image and video signals that need a high dimensional representation, often storage, processing, transmission, and analysis of such signals is a difficult task. It is therefore desirable to obtain a low-dimensional representation for such signals, and at the same time correct for corruptions, errors, and outliers, so that the signals could be readily used for later processing. Major recent advances in low-rank modelling in this context were initiated by the work of Candès *et al.* [17] where the authors provided a solution for the long-standing problem of decomposing a matrix into low-rank and sparse components in a Robust Principal Component Analysis (RPCA) framework. However, for computer vision applications RPCA is often too complex, and/or may not yield desirable results. The low-rank component obtained by the RPCA has usually an unnecessarily high rank, while in certain tasks lower dimensional representations are required. The RPCA has the ability to robustly estimate noise and outliers and separate them from the low-rank component, by a sparse part. But, it has no mechanism of providing an insight into the structure of the sparse solution, nor a way to further decompose the sparse part into a random noise and a structured sparse component that would be advantageous in many computer vision tasks. As videos signals are usually captured by a camera that is moving, obtaining a low-rank component by RPCA becomes impossible. In this thesis, novel Approximated RPCA algorithms are presented, targeting different shortcomings of the RPCA. The Approximated RPCA was analysed to identify the most time consuming RPCA solutions, and replace them with simpler yet tractable alternative solutions. The proposed method is able to obtain the exact desired rank for the low-rank component while estimating a global transformation to describe camera-induced motion. Furthermore, it is able to

decompose the sparse part into a foreground sparse component, and a random noise part that contains no useful information for computer vision processing. The foreground sparse component is obtained by several novel structured sparsity-inducing norms, that better encapsulate the needed pixel structure in visual signals. Moreover, algorithms for reducing complexity of low-rank estimation have been proposed that achieve significant complexity reduction without sacrificing the visual representation of video and image information. The proposed algorithms are applied to several fundamental computer vision tasks, namely, high efficiency video coding, batch image alignment, inpainting, and recovery, video stabilisation, background modelling and foreground segmentation, robust subspace clustering and motion estimation, face recognition, and ultra high definition image and video super-resolution. The algorithms proposed in this thesis including batch image alignment and recovery, background modelling and foreground segmentation, robust subspace clustering and motion segmentation, and ultra high definition image and video super-resolution achieve either state-of-the-art or comparable results to existing methods.

Acknowledgments

First of all, I would like to thank my supervisor Prof. Ebroul Izquierdo for his guidance throughout my PhD study and also giving me a life-altering opportunity to carry out research under his supervision; his tutelage, advice, generosity, dedication and commitment to research, boundless encouragement, and patience has and will continue to inspire me throughout life. I would also like to extend special thanks to Dr. Valia Guerra Ones from TU Delft with whom I had the privilege of working closely since the beginning of my PhD and also for countless hours of invaluable discussions, fruitful suggestions, and constructive recommendations. She has been a source of inspiration as well as mindful support for the whole duration of my PhD.

I am grateful to all my friends for enriching my life in London. Huge thanks go to all current and previous MMV members for their help and support over the years. The multiculturalism in MMV (members form 22 countries) has given me precious experiences as well as widening my world-view.

Finally, I am utmost grateful to my family, my dear father Amir, my loving mother Saeedeh, my brother Mohammad, and the latest member of the family my beautiful sister-in-law Fatemeh, for their unconditional support and encouragements through darkness and bright of the days in my life. This thesis, like all other achievements of my life, would not have been possible without them. This thesis is dedicated to them.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

In many fundamental applications in computer vision, finding a low-dimensional representation for the high dimensional data is desired. The obtained low-dimensional representation can be a basis of a certain subspace, that can be used to reduce the dimensionality or suppress noise and outliers. Example applications include but are not limited to background modelling and foreground segmentation, face detection, digit recognition, motion estimation, activity recognition, super-resolution, subspace clustering, high efficiency video coding, etc. Major recent advances in low-rank modelling in this context were initiated by the work of Candès *et al.* [17] where the authors provided a solution for the long-standing problem of decomposing a matrix into low-rank and sparse components in a Robust Principal Component Analysis (RPCA) framework. A plethora of works based on RPCA appeared in the literature that improved upon the original proposal and suggested impressive applications in computer vision tasks.

In this thesis we propose novel formulations and extensions based on low-rank and sparse decomposition for robust subspace estimation and representation. We are motivated by the fact that RPCA methods are generally computationally expensive, and the algorithms for solving those methods are unnecessarily complex. We demonstrate how a further relaxation of the RPCA solution, can work out in favour of a number of com-

puter vision tasks. Our proposed method is named Approximated RPCA (ARPCA) that has a controllable rank component that enables us to solve four fundamental computer vision problems including high efficiency video coding (HEVC), background modelling and foreground segmentation, motion subspace decomposition and clustering, and image and video super-resolution.

1.1 Overview, Motivation, and Contributions

Due to the nature of video signals, which require huge amounts of bits for storage and transmission, for more than 30 years video compression has been an active research area. New improvements in video coding have enabled UHD videos now becoming available for streaming on websites such as Youtube. However, the current standards would still benefit from further reduction of bitrate for the same quality. We present a low-rank and sparse decomposition adapted framework for HEVC, where the amount of bitrate that is used for storage and transmission could be decreased. The proposal is to use the spatio-temporal redundancy in the adjacent frames in the video sequence, to create a single background frame that can describe most of the pixel content in a group of pictures (GOP) by decomposition of the said frames into a background and several foreground frames. We show how this decomposition can drastically reduce the number of pixels that need to be encoded by HEVC, and as such can reduce the bitrate. The quality and bitrate are controllable to obtain the optimal settings for the HEVC encoder/decoder.

Applications such as face, digit, and object recognition are problem domains in computer vision where low-dimensional linear models have received a great deal of attention. The available substantial data can become very difficult to process if the difficulties such as significant illumination variation, occlusion, misalignment, deformities, and noise are not dealt with using a proper method. We propose an Approximated RPCA that can simultaneously remove shadows, occlusions, and corruptions from a set of linearly correlated images or video frames, while correcting for misalignment between them. The

obtained aligned and corrected images can then be readily used for further recognition tasks.

Background subtraction/modelling can be defined as segmentation of a video sequence into the foreground, that can contain the moving or static foreground objects in the scene, and the background, which is the static or dynamic background information in the video. It is typically used as a pre-processing step for many computer vision problems, such as automated surveillance, action recognition, intelligent environments, motion analysis, and video compression. Existing state-of-the-art algorithms have been able to address the existing challenges in background modelling and foreground segmentation to some extent; however, recent analysis of these methods reveals that there is need for a single algorithm that can tackle most or all of these challenges simultaneously. Addressing these challenges, leads to a number of considerations in designing a background model, as well as expected behavior from foreground objects, which in complex real-life applications remains an open problem. Motivated by this, we propose an Approximated RPCA method that has the following properties that are desired in a background modelling and foreground segmentation system: The sparse component structured in a novel group structure, namely a dynamic block structure and a dynamic superpixel structure that can well describe the continuity of the pixel structure of foreground objects as well as their compactness. A within-patch normalised regularisation is used to induce insensitivity to foreground object sizes, which is a common problem with RPCA-based methods. Moreover, the input video can be decomposed into an additional noise component for discarding false positive pixels (false alarms). The rank of the low-rank component is adjustable to accommodate illumination and small scene changes; per-problem tuning is also an option. To remove the ghosting effects that persist in most RPCA-based techniques a tandem algorithm is proposed, that targets the unascertained prior knowledge of distribution of outliers. To further reduce the curse of scale, a dimensionality reduction for RPCA via the column subset selection algorithm is proposed; this method eliminates the bootstrapping problem while reducing the computational complexity.

Subspace segmentation or clustering is an important fundamental task in computer vision, with applications in object tracking, motion analysis, instance segmentation, etc. The goal of subspace clustering is to segment (cluster or group) data into clusters with each cluster corresponding to a subspace. Apart from video and image pixels, we demonstrate that our Approximated RPCA can be used to decompose motion trajectories drawn from a union of multiple independent or dependent subspaces. Our method is able to remove possible errors and cluster the samples into their respective subspaces, while revealing the independent motion of each subspace when the camera is moving. The independent motion estimation for each subspace is a challenging task, as each motion subspace can be perturbed by noise and the camera motion, as well as the motion belonging to the other subspaces that might overlap with the motion subspace in question. We show that the assumption for low-rank, sparse, and noise modelling of the samples is effective, and can assist to cluster multiple subspaces in the scene. This proposal has been shown to be also effective in face clustering.

Sparse coding-based applications have been successfully applied to the single-image super-resolution (SR) problem. Conventional multi-image SR algorithms incorporate auxiliary frames into the model by a registration process using subpixel block matching algorithms that are computationally expensive. This becomes increasingly important as super-resolving UHD video content with existing sparse-based SR approaches become less efficient. In order to fully utilise the spatio-temporal information, we propose a novel multi-frame video SR approach that is aided by a low-rank plus sparse decomposition of the video sequence. We introduce a group of pictures structure where we seek a rank-1 low-rank part that recovers the shared spatio-temporal information among the frames in the GOP. Then we super-resolve the low-rank frame and sparse frames separately. This assumption results in significant time reductions, as well as surpassing state-of-the-art performance both qualitatively and quantitatively.

1.2 Notations

In this thesis the following homogenised notations are used. Notations specific to each chapter are defined locally within the corresponding chapter.

1.2.1 Data matrices

- **Matrices:** For matrices, A stands for the observation matrix, L is the low-rank matrix, S is the sparse matrix, and G (or E) is the noise matrix. For specific matrices, the notations are given in the section containing the corresponding matrices.

- **Indices:** The indices m and n are commonly used throughout this thesis to refer to the number of rows and columns of the observed data matrix A respectively. In the cases where A contains a video sequence, m is the number of pixels in each frame, and n is the number of frames. i and j are used to enumerate the individual pixels of the matrix. k is the estimated or fixed rank of the matrix L. In certain parts of this thesis if the above notations are used to refer to something other than described, it is clarified accordingly.

1.2.2 Norms

Different norms are used in this thesis for matrices.

- **Matrix ℓ_α-norm:** with $0 \leq \alpha \leq 2$, $\|M\|_0$ is the ℓ_0-norm of the matrix M, and it corresponds to the number of non-zero entries. $\|M\|_1 = \sum_{i,j} |M_{i,j}|$ is the ℓ_1-norm of the matrix M, and corresponds to the Manhattan distance. $\|M\|_2 = \sqrt{\sum_{i,j} M_{i,j}^2}$ is the ℓ_2-norm of the matrix M.

- **Matrix ℓ_∞-norm:** $\|M\|_\infty = \max_{ij} |M_{ij}|$ is the ℓ_∞-norm of the matrix M. It can be used to capture the quantisation error of the observed value of the pixel, and is equivalent to the max-norm.

- **Matrix $\ell_{\alpha,\beta}$-norm:** The structured norm with $0 \leq \alpha,\ \beta \leq 2$. $\|M\|_{\alpha,\beta}$ is the $\ell_{\alpha,\beta}$ mixed norm of the matrix M, and it corresponds to the ℓ_β-norm of the vector formed by taking the ℓ_α-norm of the columns of the underlying matrix. For instance, $\|M\|_{2,0}$ corresponds to the number of non-zero columns of the matrix M. $\|M\|_{2,1}$ induces spatial homogeneous fitting in the matrix M, and is suitable when outliers and noise are present.

- **Matrix Forbenius norm:** $\|M\|_F = \sqrt{\sum_{i,j} M_{i,j}^2}$ also known as the Euclidean norm, which should not be confused with the vector ℓ_2-norm that is also called Euclidean norm.

- **Matrix nuclear norm:** $\|M\|_*$ is the nuclear norm of the matrix M, and corresponds to the sum of its singular values. The nuclear norm is the ℓ_1-norm applied on the vector composed with the singular values of the matrix. The nuclear norm is equivalent to the Ky Fan n-norm and the Schatten-1-norm.

1.2.3 Loss functions and regularisation functions

The loss functions are used for the minimisation terms, and the regularised functions are used to enforce the low-rank, sparse, and noise constraints L, S, and G (or E), respectively. Surrogate loss functions are used in practice, in lieu of the original loss functions, to reach a solvable convex problem.

1.3 Organisation of the Thesis

The rest of this thesis is organised as follows.

Chapter 2 presents an overview of fundamental linear subspace estimation and low-rank and sparse representation concepts and related background. It also includes a detailed survey of RPCA solutions as well as their limitations.

Chapter 3 covers our modified low-rank and sparse decomposition (LRSD) for high efficiency video coding (HEVC) applications. It describes the developed techniques and models that mould the output of the LRSD to be ready for video compression standard techniques. This chapter also includes a thorough examination of efficacy of the proposed techniques.

Chapter 4 describes our Approximated RPCA framework for alignment and recovery of image and video sequences. It includes a visual validation of the proposed method, which shall be used in the upcoming chapters with certain modifications per problem.

Chapter 5 presents our Approximated RPCA framework for the task of background modelling and foreground segmentation. In this chapter we present novel techniques that enable our method to outperform rival contenders in background modelling and foreground segmentation, in four benchmark datasets.

Chapter 6 details a subspace clustering algorithm based on our Approximated RPCA framework. A comprehensive set of experiments are conducted to validate that our approach outperforms state-of-the-art methods.

Chapter 7 discusses a UHD video super-resolution method with an Approximated RPCA aided sparse representation method. It includes an overview of sparse coding-based proposals for super-resolution as well as our adaptation for super-resolving UHD video content. A thorough experimental validation and analysis is included at the end of this chapter.

Chapter 8 concludes this thesis by providing conclusions and observations on the proposed methods. It also includes some directions and ideas for future developments of the proposed algorithms.

Chapter 2

Background on Subspace Estimation

The research presented in this thesis follows recent developments in the field of *Robust Principal Component Analysis* (RPCA). A critical breakthrough in this context was reported by Candès *et al.* [17] where the authors provided a solution for the long-standing problem of decomposing a matrix A into two components such that $A = L + S$, where L is a low-rank matrix and S is a sparse matrix. Essentially, A contains the training sequence.

As a simple example, RPCA can be applied to separate moving objects from a static or moving background. This is a basic video processing task with manifold applications including automated anomaly detection in video surveillance, face alignment for recognition and authentication, human motion analysis, action recognition, object tracking, video summarisation retrieval and editing, and object based video coding. This task is commonly referred to as foreground/background separation in the literature – where foreground represents the moving or static objects of interest in the scene, and the background represents the mostly static parts of the scene that may not be useful for further processing. Indeed, many algorithms and techniques have been developed since the early

days of digital image processing with varied degrees of performance. With RPCA in a foreground/background separation case, the background sequence is modelled by the low-rank subspace L that can gradually change over time, while the moving foreground objects constitute the correlated sparse outliers S.

Candès *et al.* [17] showed that, under certain trivial assumptions, the decomposition problem can be solved by means of a convex program referred to as *Principal Component Pursuit* (PCP). The solutions of this optimisation problem are a low-rank part (representing the static samples in the scene, namely the "background"), and a sparse noise part (representing the foreground or moving objects in the scene plus additive noise).

These advancements in RPCA are fundamental and have been applied to background modelling and foreground detection, as well as removing shadows and specularities in images of faces. However PCP imposes a number of limitations. In unconstrained real-world video sequences background usually contains objects that do not contribute to the foreground information in the scene. Background objects can be stationary, such as walls, doors, furniture, or non-stationary such as waving trees, rippling water surface, or moving escalators. The appearance of the background and objects belonging to background often undergoes various changes over time, e.g. changes in brightness caused by weather conditions or light switches in indoor scenes. Older video segmentation methods are mostly constrained to stationary backgrounds or backgrounds undergoing small camera jitter with a minor change only in camera perspective. However in real-world applications the camera motion can be very large.

The approach presented in this thesis tries to fill a gap in the related literature, to adapt the RPCA to a number of computer vision applications; emphasis is put on the concept that a given video sequence or a set of linearly-correlated images can be decomposed into a high-frequency (sparse) and a low-frequency (low-rank) component. We dare to call the low-rank component the *background* and the sparse part the *foreground* of a given video sequence or a set of images; the intuition behind this is that it is expected that following this methodology, most of the extracted objects or regions

that are of interest for later processing, would lie in the sparse domain. This assumption opens up interesting solutions for some long-standing computer vision tasks. The adaptations of RPCA to computer vision tasks presented in this thesis include, but are not limited to simultaneously addressing critical aspects such as: (1) handling camera movement, (2) treating the pixel structure in the sparse matrix in blocks, (3) removing an effect called ghosting where the foreground is absorbed into background, (4) and finally calculating the minimisation problem with a computationally cheap algorithm.

In the following sections we briefly introduce the linear subspace estimation, and its several applications in computer vision.

2.1 Linear Subspace Estimation

In 1999, Oliver *et al.* [120] were the first authors to model the background by *Principal Component Analysis* (PCA). They developed the theory of *Robust Subspace Learning* (RSL) for making linear learning methods robust to outliers (contaminations in signals) which are common in realistic training sets. As a simple example consider a set of 2-dimensional points drawn from a random distribution, as shown in Figure 2.1. Principal Component Analysis (PCA) is the most popular method to estimate an orthogonal basis set, where a set of orthonormal eigenvectors and their corresponding eigenvalues are calculated such that the reconstruction error by re-projection along those directions is minimum. Using the corresponding eigenvalues, these bases can be sorted according to the variance along each basis direction. The insignificant basis vectors (with the smallest variance) are ignored in practice, as they typically correspond to noise. As a result, a major reduction of dimensionality is obtained which is desirable in machine learning techniques that scale exponentially with dimension of data. The calculated basis vectors are illustrated in Figure 2.1, where the length of each vector is weighted using the eigenvalues.

This can be applied in a wide variety of machine learning and computer vision appli-

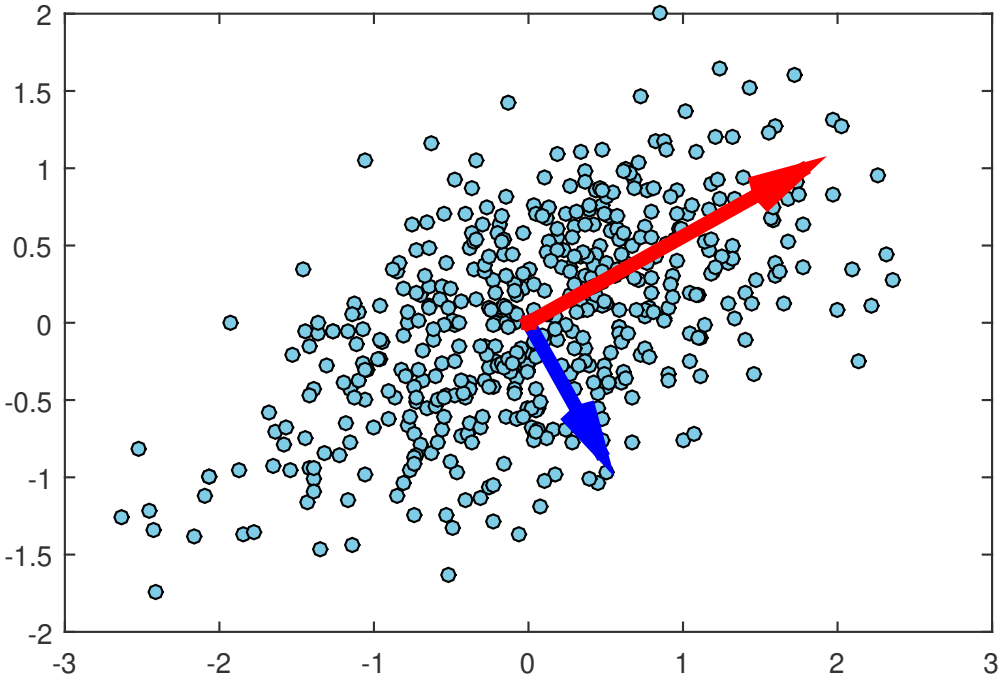


Figure 2.1: Sample data points drawn from a random distribution. The arrows correspond to the estimated basis vectors computed using PCA. The length of each basis vector corresponds to the amount of data variance along the direction shown.

cations. For instance, consider a set of images of size 100×100 of a human face under different lighting conditions, poses, etc. To represent each image 10000 dimensions are required. By applying PCA, the major k basis is extracted. That means, each image can be represented as only a set of k coefficients. Those coefficients provide a representation of the faces that can be used to classify the face images.

Many extensions for the PCA have been proposed in the literature. We briefly discuss a number of most prominent ones.

- Kernel PCA (KPCA) [135]: Extends PCA to handle data lying in a non-linear subspace. As the data is embedded in a higher dimensional space, it can be linearly separated. Instead of finding this non-linear mapping, each point is represented by the distances to all other points that form a kernel matrix, and Eigen Value Decomposition is applied on the new representation. Because the actual high dimensional

embedding is not explicitly computed, the kernel PCA does not compute the principal components themselves, but instead the projections of the data onto those components.

- Probabilistic PCA [150]: PCA is formulated as a Maximum Likelihood parameter estimation problem. Expectation Maximisation (EM) is used to compute the principal subspace in an iterative process. Probabilistic PCA can handle missing values, as it estimates a generative model.

- Exponential family generalisation of PCA [25]: PCA assumes a squared loss function. This extension enables PCA to handle loss functions that are better suited for several data types, such as non real-valued, binary, integer, or non-negative data.

- Generalised PCA (GPCA) [156]: Extends PCA to handle data points drawn from multiple subspaces.

A critical issue with classical PCA is sensitivity to outliers, that result in potentially inaccurate basis computation. This is the most important reason behind research for a method that is able to detect outliers simultaneously with estimating the basis. In the next section, we discuss the recent developments in low-rank and sparse optimisation, that is one of the most successful approaches for robust subspace estimation.

2.2 Low-Rank and Sparse Representation

We can define a low-rank structure to be a set of observations represented by a low number of bases. Such structures can be seen in many natural images that contain repetitive structure compounds such as a building facade. Similarly, the frames in a video sequence could also form a low-rank subspace, as the adjacent frames are typically highly correlated. Figure 2.2 shows various examples in computer vision, where a low-rank structure could be found.

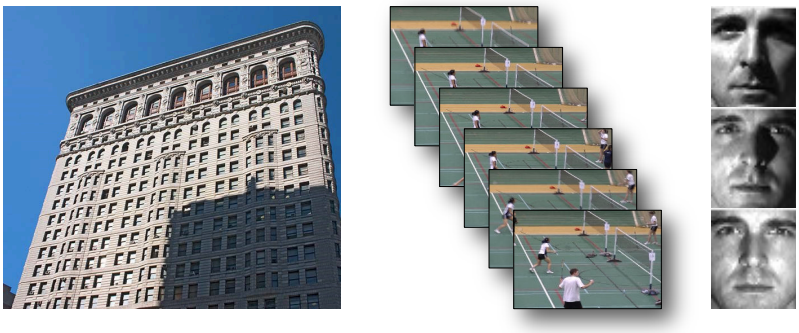


Figure 2.2: Example low-rank structures. Left: building facade. Middle: video sequence frames. Right: images of human face under different illuminations.

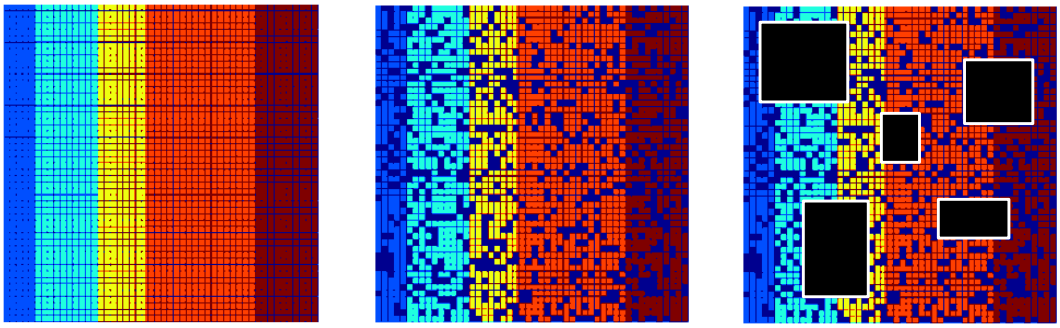


Figure 2.3: Illustration of different types of matrix corruptions. Left: original data matrix. Middle: element-wise corruptions. Right: both element-wise corruptions and missing data. This picture is taken from [106].

These observed low-rank structures, are often grossly corrupted by outliers. These outliers can exhibit as element-wise noise or corruptions, row- or column-wise corruptions, or missing data due to an acquisition problem. These corruptions are difficult to model. Figure 2.3 shows each case. As a real-world example, the building facade in Figure 2.4 is occluded by two structurally different images. Consequently, the occluded pixels do not lie in a low-rank space of the building facade. In these examples, it is desirable to separate the noise, which can itself be of interest in specific applications, by recovering the inherent low-rank structure.

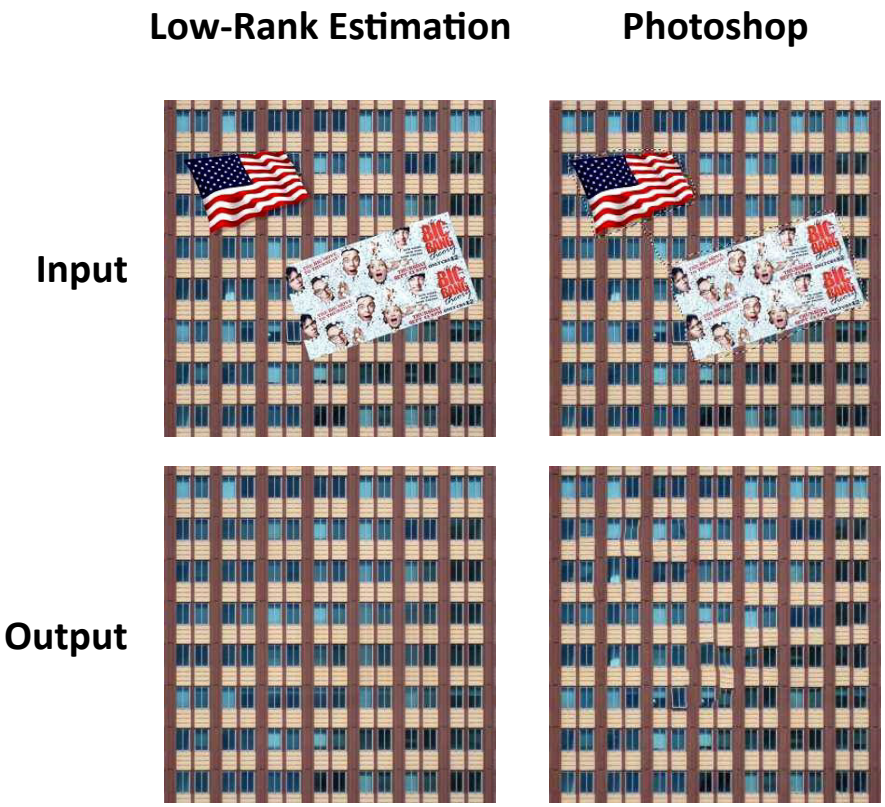


Figure 2.4: Example of an occluded image of a building facade. The rank of the matrix containing this image is unnecessarily high due to the occlusion. Low-rank optimisation makes it possible to detect the noise (the occlusion) and thus recover the original low-rank structure (the facade image). The obtained low-rank facade preserves the architectural symmetry of the windows better where the images was occluded by the flag and the poster. This picture is taken from [107].

One can stack the noisy data points as the columns of a matrix, and decompose the obtained matrix into a low-rank component and a sparse noise component by minimising the nuclear norm and the ℓ_1-norm, respectively. Assume a set of frames from a video sequence shown in Figure 2.5. To eliminate spatial dependency, each frame is vectorised as a column in a new matrix that can be readily decomposed as described.

The assumption of decomposability of a set of linearly-correlated images (or video frames) provides an excellent potential for many image and video processing tasks. The low-rank and sparse decomposition can be employed for background subtraction and

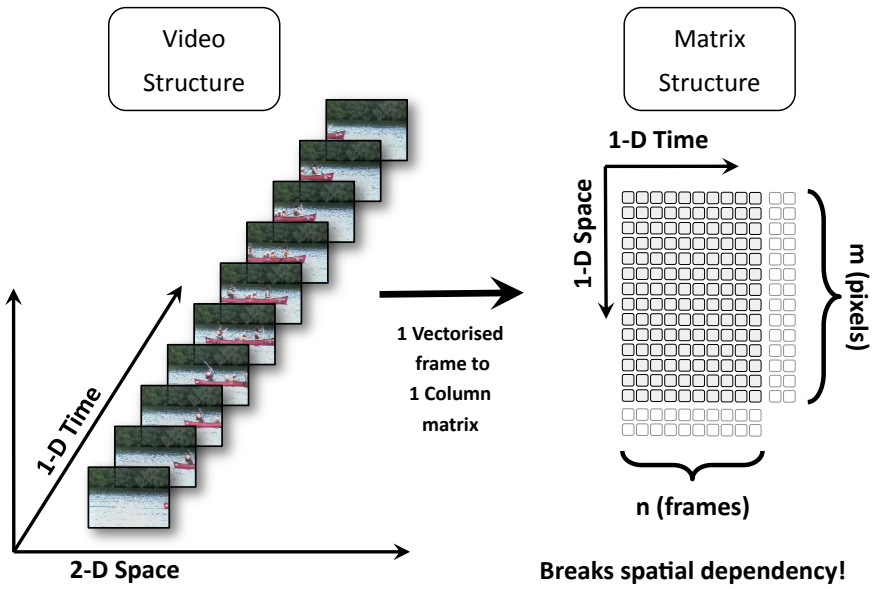


Figure 2.5: Data structure transformation.

Figure 2.6: Background subtraction and foreground segmentation by decomposition of the video sequence into a low-rank background sequence, and a sparse foreground sequence.

foreground segmentation in video sequences as shown in Figure 2.6. The original video sequence can be decomposed into a low-rank component containing the highly-correlated pixels that correspond to the mostly-unchanging background pixels, and a sparse component containing sparse noise that could correspond to the foreground objects.

Although there have been several improvements for PCA [151] that addressed the limitations of classical PCA with respect to outlier and noise – yielding the field of robust PCA – these methods may not achieve sufficient performance for applications such as surveillance that require fast and very accurate computation. The first works on robust PCA (RPCA) developed by [17], and [163], proposed convex optimisation strategies

by which a low-rank matrix was recovered from a small set of corrupted observations. They showed that RPCA can be solved under broad conditions via convex optimisation techniques such as the Principal Component Pursuit (PCP), that recovers the low-rank and the sparse matrices.

Historically, the RPCA problem has been solved via popular convex optimisation techniques such as the Augmented Lagrange Multiplier (ALM) [95] and the Accelerated Proximal Gradient (APG) [96] under broad conditions, and is widely used in large-scale problems such as in [149], [174], [62], [122], [164], [121], [124], [136], [101], and [58]. These approaches are based on the Alternating Directions Method of Multipliers (ADMM), that was introduced by authors of [57], [59].

Given a large data matrix A of size $m \times n$, (typically $m \gg n$) and assuming that A has rank $k \leq n$, the decomposition by RPCA is defined as $A = L + S$ with

$$\min \|A - L\| \quad \text{subject to} \quad rank(L) \leq k, \tag{2.1}$$

where L is low-rank and S is sparse. The straightforward formulation of the proposed method is to use ℓ_0-norm to minimise the energy function

$$\arg\min Rank(L) + \lambda \|S\|_0 \quad \text{subject to} \quad A = L + S, \tag{2.2}$$

in which λ is an arbitrary balanced parameter. The above problem is NP-hard and thus infeasible for practical applications. To provide a feasible solution the authors in [17] proposed to minimise a surrogate model using $\lambda = \frac{1}{\sqrt{\max(m,n)}}$ and the ℓ_1 and nuclear norms instead. This leads to the convex problem

$$\arg\min \|L\|_* + \lambda \|S\|_1 \quad \text{subject to} \quad A = L + S \tag{2.3}$$

Unfortunately this method may be too complex for a practical usage in some appli-

cations. RPCA offers a blind low-rank and sparse noise separation; in other words, it is assumed that the rank of the low-rank component is not too large and the sparse component is reasonably sparse (uniformly random). RPCA is also limited to the low-rank component being exactly low-rank and the sparse component being exactly sparse. However in realistic and unconstrained video footages these assumptions are not always satisfied. It is worthwhile investigating when either or both these assumptions are relaxed, as in some applications the obtained low-rank component would still have an unnecessarily high rank. Another important aspect is to further investigate developing algorithms that have better scalability. It would specifically be useful to scenarios where we have hours of video sequences.

In order to reduce complexity of the algorithm, the GoDec method was proposed by [176]. The method had the goal of decomposing a matrix into low-rank and sparse components in noisy case. They estimate the low-rank part L and the sparse part S of a matrix A with noise part E.

$$A = L + S + E \tag{2.4}$$

GoDec alternatively assigns the low-rank approximation of $A - S$ to L and the sparse approximation of $A - L$ to S. The authors also proved that the objective value $\|A - L - S\|_F^2$ converges to a local minimum, while L and S linearly converge to local optimums. The optimisation problem becomes

$$\min_{L,S} \|A - L - S\|_F^2 \quad \text{such that} \quad rank(L) \leq k, card(S) \leq \kappa \tag{2.5}$$

The noisy model $A = L + S + E$, can handle approximated decomposition even when the exact and unique RPCA decomposition does not exist, due to additive noise. In the optimisation process the rank of L and cardinality of S are fixed which imposes limitations to decomposition of unconstrained-environment video sequences, as the amount of information needed to reconstruct a video sequence compared to another varies and

therefore at least either the rank of L or the cardinality of S must be flexible. Moreover, the hard-thresholding towards S requires sorting all its entries' magnitudes and thus is time consuming. Later a similar method was proposed [177] with introduction of a *Lagrange* formulation.

$$\min_{L,S} \|A - L - S\|_F^2 + \lambda\|S\|_1 \quad \text{such that} \quad rank(L) \leq k \tag{2.6}$$

Tuning *Lagrangian* parameter which acts as a soft-threshold value is much more convenient than determining the cardinality of S, because the resulting decomposition error is more robust to the change of the *Lagrangian* parameter. To solve (2.6) they decompose the sparse part as the sum of several low-rank matrices that each correspond to objects in the scene sharing the same motion trajectory. However this method does not handle cases with moving cameras (parts of the scene moving uniformly with the camera motion) or cases where part of the background has the same motion trajectory as the foreground such as people on an escalator.

All these approaches generally perform at their best under particular conditions: the sequence should be comprised of a set of images that are not misaligned, i.e., the frames must be taken with a fixed camera without jitter or movement. Furthermore, the illumination in the scene should be constant, and the background must remain static and no objects should be introduced into it. In practice these conditions are rarely met.

Furthermore, matrix computations such as *Eigen Decomposition*, *QR Decomposition*, *Singular Value Decomposition* (SVD), *least squares minimisation* etc. are very computationally expensive and require huge memory. They are not well suited for data that has missing or noisy entries, is large, or needs to be processed in real-time.

The problem with most RPCA based methods is that they do not take into account the spatial connectivity of the foreground pixels that are assumed to be the sparse matrix entries. This problem appears as some foreground pixels get absorbed into the

background, so the foreground does not appear as a connected solid region. PCA provides a robust model of the probability distribution function, but not of the moving objects while they do not have a significant contribution to the model [67]. The limitations arising with this problem are firstly the size of the foreground object must be small and not appear in the same location for a long period of time; and secondly that the outliers of the foreground objects may be absorbed into the background mode without a mechanism of robust analysis [11].

Recent advances on subspace estimation by sparse representation and rank minimisation show a nice framework to separate moving objects from the background. These advances concern robust subspace tracking and RPCA models. A recent comprehensive survey of RPCA approaches can be found in [13]. For RPCA via low-rank and sparse matrix decomposition several approaches have been developed and can be classified as follows.

- RPCA via Principal Component Pursuit (RPCA-PCP) [17], [163], [19].
- RPCA via Outlier Pursuit (RPCA-OP) [166].
- RPCA via Sparsity Control (RPCA-SpaCtrl) [113], [114].
- RPCA via Sparse Corruptions (RPCA-SpaCorr) [75].
- RPCA via Log-Sum Heuristic Recovery (RPCA-LHR) [29].
- RPCA via Iteratively Reweighted Least Squares (RPCA-IRLS) [68], [68].
- RPCA via Stochastic Optimisation (RPCA-SO) [60].
- RPCA via Dynamic Mode Decomposition (RPCA-DMD) [63].
- Bayesian RPCA (BRPCA) [30].
- Approximated RPCA (ARPCA) [176], [177], [39], [40], [41], [42], [45], [47].

- Sparse Additive Matrix Factorisation (SAMF) [116].
- Variational Bayesian Sparse Estimator (VBSE) [23].

2.2.1 Limitations of RPCA-based methods

The RPCA can be solved under minimal assumptions to recover the low-rank and the sparse matrices [17] with encouraging performance. However, for practical applications there are several limitations to this method that inhibit its deployment and integration into systems. The following limitations are the core motivations for the research presented in this thesis.

1- The algorithms to solve RPCA are computationally expensive (e.g. Singular Value Decomposition). Despite the promising performance of RPCA, it lacks the attraction for applications such as video surveillance, medical imaging, video coding, etc., where real-time or fast performance is required.

2- RPCA solvers are mostly batch methods in which all the training frames are stacked in the input matrix. With long video sequences with high resolutions, the matrix becomes humongous, and that in turn slows down the computation heavily; moreover, eventually these matrices get so large that one would run out of memory to hold them. In most computer vision applications it would be more useful to estimate the low-rank and the sparse matrices in an incremental way for each new frame, rather than as a whole batch.

3- The spatial and temporal features are lost, as each frame is represented as a column vector in the input matrix. There is need to incorporate the underlying spatial information in the data matrix to obtain better performance in computer vision applications. Where time-domain analysis impacts the performance, a mechanism must be devised to handle both spatial and temporal information.

4- The classical PCP solution of RPCA imposes the low-rank component being exactly

low-rank and the sparse component being exactly sparse but in real-world applications such as video surveillance, the data are often corrupted by noise affecting every entry of the data matrix. Moreover, the matrix containing the data can exhibit low-rank and sparse properties simultaneously. There is need for devising more intelligent norms to handle spatial contiguity in the sparse matrix, as well as flexibility in the low-rank component.

5- Last but not least, RPCA-PCP assumes that all entries of the matrix to be recovered are exactly known via the observation and that the distribution of corruption should be sparse and random enough without noise.

In addition to the above limitations, in real applications only a fraction of the data can be observed in some environments. Moreover, the observations could be corrupted by noise, and the foreground moving objects (assumed to reside in the sparse matrix) are spatially localised and grouped in the matrix containing the image. More concretely, in the low-rank and sparse decomposition, the outliers and noise are assumed as sparsity patterns and are uniformly scattered. However, this is not realistic when dealing with images and videos. The moving objects usually present themselves as a connected region of pixels in the data matrix, continuously in temporal domain. This then enforces the need for a proper spatio-temporal solution.

2.2.2 RPCA methods solved via PCP

Many efforts have been made to develop low-computation algorithms for solving PCP. Incremental algorithms and real-time implementations of PCP have been proposed to update the low-rank and sparse matrix when a new data arrives. Other efforts have addressed problems that appear specifically in real application such as: presence of noise, quantisation of the pixels, spatial constraints of the foreground pixels, and local variations in the background. To address presence of noise, [179] proposed a stable PCP (SPCP) that guarantees stable and accurate recovery in the presence of entry-wise noise.

Method	Decomposition	Minimisation	Constraints
PCP [17]	$A = L + S$	$\min_{L,S} \|L\|_* + \lambda\|S\|_1$	$A - L - S = 0$
SPCP [179]	$A = L + S + E$	$\min_{L,S} \|L\|_* + \lambda\|S\|_1$	$\|A - L - S\|_F < \delta$
QPCP [7]	$A = L + S$	$\min_{L,S} \|L\|_* + \lambda\|S\|_1$	$\|A - L - S\|_\infty < 0.5$
BPCP [148]	$A = L + S$	$\min_{L,S} \|L\|_* + \kappa(1-\lambda)\|L\|_{2,1} + \kappa\lambda\|S\|_{2,1}$	$A - L - S = 0$
LPCP [162]	$A = AU + S$	$\min_{U,S} \alpha\|U\|_* + \beta\|U\|_{2,1} + \beta\|S\|_1$	$A - AU + S = 0$

Table 2-A: Convex RPCA via PCP models.

An inequality constrained version of PCP was proposed in [7] to take into account the quantisation error of the pixel values. A block-based PCP (BPCP) method was proposed by [148] that acts via a decomposition that enforces the low-rankness of one part and the block-sparsity of the other. Another work [162] used a decomposition corresponding to a more general underlying model consisting of a union of low-dimensional subspaces for local variation in the background. Table 2-A shows an overview of a handful of the different versions of PRCA-PCP in term of minimisation, constraints, and convexity. For a complete list of the RPCA solvers refer to a recent survey by [13].

The recent advances in RPCA via PCP are fundamental and can be applied to a number of computer vision tasks. However no algorithm as of yet seems to emerge that is able to simultaneously address all the key challenges that accompany real-world videos. This is due, in part, to the absence of a rigorous quantitative evaluation with synthetic and realistic large-scale dataset with accurate ground truth providing a balanced coverage of the range of challenges present in the real world situations [10], [11], [13].

In the recent literature several algorithms with different complexity and contributions have been proposed to solve the PCP problem [12]. As an example, these algorithms involve solving the minimisation problems that appear in table 2-A in each iteration. Those minimisation problems can have a closed-form solution or not, depending on the application. If a closed form-solution exists, PCP can be reformulated as a semi-definite program and then solved by standard interior point methods. However, interior point methods have difficulty in handling large matrices because the complexity of computing the step direction is $O((mn\min(m,n))^2)$, where $m \times n$ is the size of the data matrix.

If $m = n$, then the complexity is $O(n^6)$. So the generic interior point solvers are too limited for many real applications where the size of data are very large. To overcome this scalability issue, only the first-order information can be used. In [16] it was shown that this technique, called Singular Value Thresholding (SVT), can be used to minimise the nuclear norm for matrix completion. As the matrix recovery problem in equation (2.3) needs minimising a combination of both the ℓ_1-norm and the nuclear norm, [163] adopted an Iterative Thresholding technique (IT) to solve it and obtained similar convergence and scalability properties to the interior point methods. However, the Iterative Thresholding scheme converges extremely slowly. To alleviate this slow convergence, two algorithms were proposed in [96]: the accelerated proximal gradient (APG) algorithm and the gradient-ascent algorithm applied to the dual form of the problem in equation (2.3). However, these algorithms are all the same too slow for real applications. More recently, [95] proposed two algorithms based on Augmented Lagrange Multipliers (ALM). The first algorithm is called Exact ALM (EALM) method that has a Q-linear convergence speed, while the APG is in theory only sub-linear. The second algorithm is an improvement of the EALM that is called Inexact ALM (IALM) method, which converges practically as fast as the EALM, but the required number of partial SVDs is significantly less. The IALM is at least five times faster than APG, and its precision is also higher [95]. However, the direct application of ALM treats the equation (2.3) as a generic minimisation problem and ignores its separable structure emerging in both the objective function and the constraint [95]. Hence, the variables S and L are minimised simultaneously. Authors in [171] proposed to alleviate this ignorance by the Alternating Direction Method (ADM) which minimises the variables L and S serially. ADM achieves it with less computation cost than ALM. Recently, [20] proposed a non-convex splitting version of the ADM [95] called NCSADM. This non-convex generalisation of [95] produces a sparser model that is better able to separate moving objects and stationary objects. Furthermore, this splitting algorithm maintains the background model while removing substantial noise, more so than the convex regularisation does.

In the second case when the resulting sub-problems do not have closed-form solutions, [169] proposed to linearise these sub-problems such that closed-form solutions of these linearised sub-problems can be easily derived. Global convergence of these Linearised ALM (LALM) and ADM (LADM) algorithms are established under standard assumptions. Recently, [100] improved the convergence for the Linearised Alternating Direction Method with an Adaptive Penalty (LADMAP). They proved the global convergence of LADM and applied it to solve low-rank representation (LRR). Furthermore, the fast version of LADMAP reduces the complexity $O(mn\min(m,n))$ of the original LADM based method to $O(rmn)$ where r is the rank of the matrix to recover, which is supposed to be smaller than m and n. In a similar way, [105] and [62] proposed a Linearised Symmetric Alternating Direction Method (LSADM) for minimising the sum of two convex functions. This method requires at most $O(\frac{1}{\epsilon})$ iterations to obtain an ϵ-optimal solution, and its fast version called Fast-LSADM requires at most $O(\frac{1}{\sqrt{\epsilon}})$ with little change in the computational effort required at each iteration.

All these methods require computing SVDs for some matrices, resulting in complexity of $O((mn\min(m,n)))$. Although partial SVDs are used to reduce the complexity to $O(rmn)$ such a complexity is still high for large datasets. Therefore, recent researches focus on the reduction of the complexity by avoiding computation of SVD. Another work [136] presented a method where the low-rank matrix is decomposed in a product of two low-rank matrices and then minimised over the two matrices alternatively. Although, they do not require nuclear norm minimisation and so the computation of SVD, the convergence of the algorithm is not guaranteed as the problem is non-convex. Furthermore, both the matrix multiplication and QR decomposition based rank estimation technique require $O(rmn)$ complexity. So, this method does not essentially reduce the complexity. In another way, [115] reduced the problem scale by random projections, but different random projections may lead to radically different results. Furthermore, additional constraint to the problem slows down the convergence. The complexity of this method is also $O(pmn)$ where $p \times m$ is the size of the random projection matrix. So, this method

still does not have linear complexity with respect to the matrix size. Recently, [100] proposed a novel algorithm, called ℓ_1-filtering for exactly solving PCP with an $O(r^2(m+n))$ complexity. This method is a truly linear cost method to solve PCP problem when the data size is very large while the target rank is small. Moreover, ℓ_1-filtering is highly parallelisable. It is the first algorithm that can exactly solve a nuclear norm minimisation problem in linear time. Numerical experiments [100] show the great performance of ℓ_1-filtering in speed compared to the previous algorithms for solving PCP. For a thorough discussion of RPCA-PCP solutions refer to [11], [12], [13].

2.2.3 Finding the optimal hyper-parameters in PCP

PCP recovers the true underlying low-rank matrix when a large partition of the measured matrix is either missing or arbitrarily corrupted. However, in the absence of a true underlying signal L and the deviation S, it is not clear how to choose a value of λ in equation (2.3) that produces a good approximation of the given data A for a given application. A typical approach would involve some cross-validation step to select λ to maximise the final results of the application. The issue with cross-validation in this situation is that the best model is selected indirectly in terms of the final results, which can depend on unexpected ways on later stages in the data processing chain of the application [12], [11]. Experimental results show that in general for the specific task of background/foreground segmentation in video sequences, the best λ is not the one determined by the theory in Candès *et al.* [17]. In the upcoming chapters we shall discuss the importance of tuning hyper-parameters, and describe our method for choosing the best λ.

Chapter 3

Low-rank and Sparse Decomposition for HEVC

High Efficiency Video Coding (HEVC) is a video compression standard approved in 2013 that is said to double the data compression ratio compared to the predecessors H.264/MPEG-4 AVC at the same level of video quality [145], [146], [78], and [161]. However, the video transmission in high resolution requires new improvements. The main objective of this chapter is to propose a new technique improving the data compression-quality rate of HEVC. A partitioning of the video into Groups of p (consecutive) Pictures (GOP) typically $p = 8$ or 16 is assumed. Usually the consecutive frames in a GOP have many pixels with similar intensities (corresponding to the background of the scene) and only a small portion of them is different (corresponding to the changing foreground). In each GOP, we consider the matrix A whose p columns are formed by concatenating the $w \times h$ pixels of the p frames. Ideally we would want to decompose the matrix A as the sum of a rank-1 matrix L plus a sparse matrix S plus an error matrix G. The matrices L and S describe the background and foreground, respectively. It means that the GOP can be recovered (except the error matrix G) using only one dense $w \times h$ vector and p sparse $w \times h$ vectors. If the number of zero elements (sparsity) of the sparse

vectors is large enough then the matrices L and S permit to encode the GOP in an efficient way. If the differences between pixel intensities of the adjacent frames in the GOP result from camera-induced motion then the low-rank and sparse decomposition (LRSD) is extended and the reconstruction of the GOP is performed using an additional transformation matrix that can describe the general background motion caused by the camera movement. The transformation matrix can be described by a vector of 6 elements. The main challenge the proposed technique is faced with is to achieve better compression-rate/quality than the rival HEVC.

3.1 Introduction

In section 2.2 we introduced many algorithms have been reported that are able to decompose a matrix as the sum of a low-rank and a sparse matrix. For a video coding application, we would be interested in algorithms where we can obtain a rank-1 approximation for the low-rank part, as this minimises the amount of data to be encoded, and transmitted. Therefore for each GOP we should obtain a rank-1 matrix that describes the unchanging background pixels of that GOP. Similar to (2.6), we can achieve this using the proposed approximated RPCA as

$$\min_{L,S} \|A - L - S\|_F^2 + \lambda \|S\|_1 \quad s.t. \quad rank(L) = 1 \tag{3.1}$$

The HEVC processes each frame of a video sequence in Coding Tree Units (CTU) quadtree structure. This is a block-based process in which the image is divided into square or rectangular blocks with pre-defined number of pixels within each block. The LRSD acts on frames of a video sequence that are stacked as columns in a huge matrix. Instead, to comply with the CTU structure in HEVC, it should be adapted to work in blocks of dimension $b \times b$ where b can vary from 16 to 64 pixels.

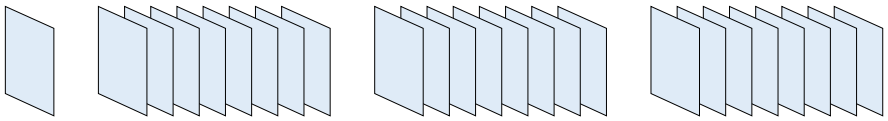


Figure 3.1: Frame grouping in HEVC encoder.

3.2 A Description of HEVC Codebase

HEVC encodes a sequence of frames by partitioning it in GOPs. Typically the GOP size is set to 8. Frames in a GOP can be encoded in any order, following a pre-defined structure which is passed to the encoder in the form of a "GOP table". The GOP table specifies, among other things:

- In which order the frames in a GOP are encoded
- Which reference frames to use for each frame in a certain position in the GOP

Due to the fact that frames in a GOP can be encoded in a different order than the temporal order at which they are displayed, HEVC allows for bi-directional motion estimation: a frame can be predicted using information extracted from a frame in the future [146]. For this reason, the encoder must have all the frames in a GOP available before it starts encoding that GOP. Notice that the first frame in the entire sequence is treated differently from all the others: it is encoded as an intra-picture, independently from all other frames, and it does not belong to any GOP. In other words a typical HEVC encoder encodes frames as depicted in Figure 3.1.

Each frame is then partitioned in blocks which are independently predicted and transformed. HEVC allows frames to be flexibly partitioned to adapt to local characteristics of the content currently being encoded. In particular, a frame is first divided in a number of Coding Tree Units (CTUs) of fixed size. CTUs must span a square region ranging from a maximum of 64×64 to a minimum of 16×16 luma samples. Each CTU is then partitioned in Coding Units (CUs) following a recursive quadtree structure. CUs are assigned a depth, depending on their size and the corresponding level of recursion. In

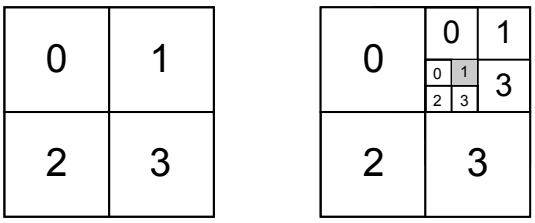


Figure 3.2: Two possible partitioning configurations for a 64×64 CTU into smaller CUs.

case a single CU is considered whose size is the same as the original CTU, the CU is assigned a depth 0. Each CU at depth d can then be partitioned into four CUs at depth $d + 1$ with half the height and width of their parent CU. This process can be repeated recursively, up to a minimum CU size of 8×8 luma samples. This means that a maximum depth of 3 is allowed in case the CTU size is set to its maximum value of 64×64 luma samples, as will be assumed in the rest of this chapter unless otherwise specified. Figure 3.2 shows two possible partitionings of a 64×64 CTU into CUs of different sizes.

The content of each CU can then be predicted following a variety of different modes, using inter or intra-prediction; a detailed description of HEVC prediction modes can be found in [145], [146], [78], and [161], and goes out of the scope of this thesis.

3.2.1 Block-GOP LRSD

A low-rank and sparse decomposition (LRSD) can be used in the context of video coding to exploit redundancy among frames in a sequence. In order to apply the best decomposition for the purpose of video coding, a specific version of LRSD was developed which we call here Block-GOP LRSD. When using Block-GOP LRSD, the decomposition is performed only on frames belonging to the same GOP. The low-rank component of a GOP is extracted using the decomposition; we refer to this component as the "background" (even though this is not an accurate nomenclature). Conversely, the sparse part is considered from all frames in the GOP, referred to as the "foreground". The decomposition is performed in such a way to maximise the number of blocks in the fore-

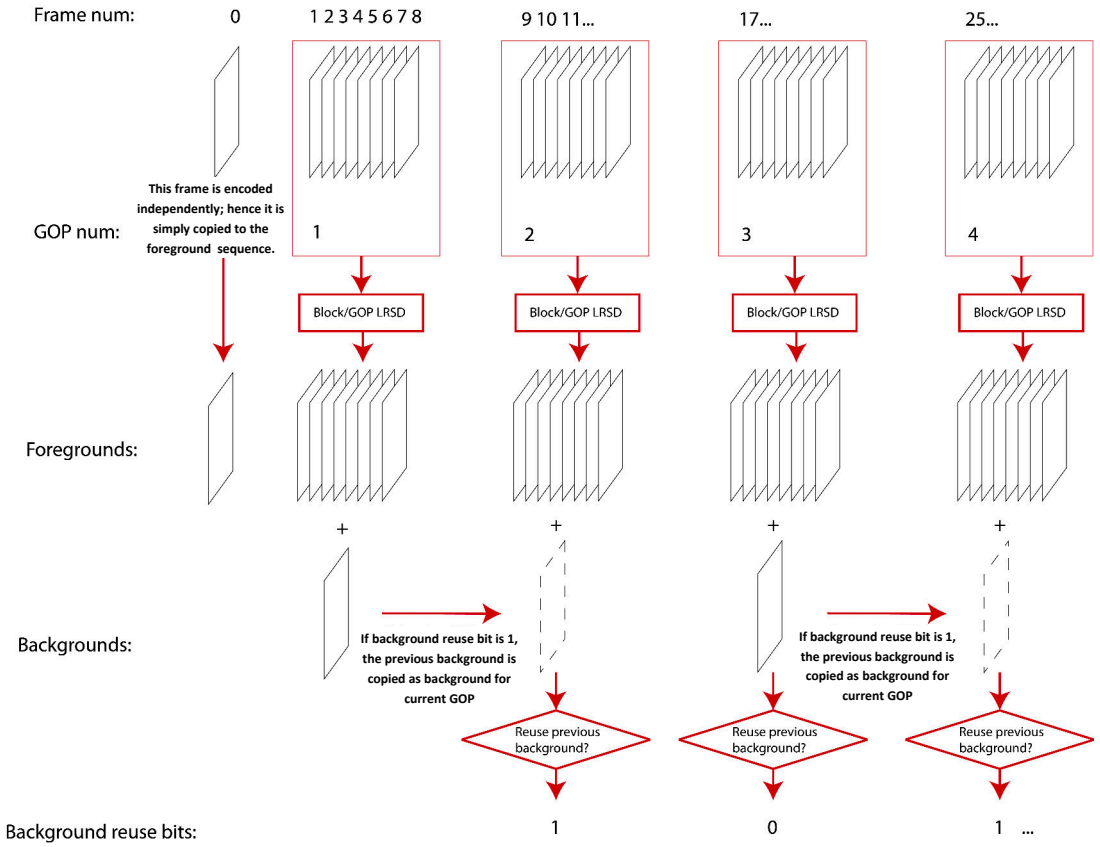


Figure 3.3: Block-GOP LRSD framework.

ground which do not have any non-zero elements. Blocks can be assigned an arbitrary size from configuration; block sizes of 8×8 or 16×16 pixels were considered in the rest of this report. A "refinement" option is also available, which considers blocks of half the height and width of the original block size: for instance in case of a block size of 16×16, blocks of 8×8 are also considered when using the refinement option. After the decomposition is performed on a given GOP, the decomposer has the option whether to use the obtained background, or whether to re-use the background which was found in the previous GOP. This is illustrated in Figure 3.3.

The output of the Block-GOP LRSD decomposition is finally the following:

- A foreground sequence. This contains: the first frame in the sequence as it is. Then, for each GOP, it contains the "sparse" component. In particular, if a block was

found with no non-zero elements, then such block of zeros is put in the foreground; if a block was found with non-zero elements, then the corresponding block in the original sequence is copied to the foreground. As a result, the foreground sequence will contain entire blocks of original content and entire blocks of zeros.

- A background sequence. This contains the background for the first GOP. Then, it contains the background for each subsequent GOP which does not reuse the background of the previous GOP. For instance in the Figure 3.3, the background sequence will contain the background of GOP 1, and the background of GOP 3.

- A sequence of bits, referred to as "background reuse bits": this is a bit for each GOP (apart from the first GOP), which signals whether a new background is needed or the previous can be used. For instance, the background reuse bit for GOP 2 is 1 in the figure, because a new background is needed; conversely, the background reuse bit for GOP 3 is 0, because the previous background can be used.

3.2.2 The LRSD-HEVC codebase

The three inputs obtained by the Block-GOP LRSD (the foreground sequence, background sequence and background reuse bits) are fed to the LRSD-HEVC codebase to perform the actual HEVC encoding. The LRSD-HEVC encoder starts with encoding the first frame in the foreground sequence; this happens as in conventional HEVC using intra-frame coding.

The encoder considers the GOP number 1. Then the following is performed:

1. The first background in the background sequence is encoded. Such encoding can be performed in two ways, selectable from configuration. In the first method, the first background is encoded as an intra-frame. This means that it is encoded without reference to any other frame; the advantage is that the quality of the reconstruction is possibly higher; the disadvantage is that intra-coding requires a huge amount of

bits. If the second method is used instead, then the first background is encoded as an inter-predicted frame, using the first frame in the foreground as reference. While it is likely that the content of these two frames is different, we can expect some similarities and hence inter-coding should drastically reduce the bits needed to encode the first background. This of course comes at the cost of some quality losses. The compressed first background frame is stored in the bitstream, whereas the reconstructed first background frame is kept in memory, ready to be used when encoding the foreground sequence.

2. The first GOP in the foreground sequence is then encoded. The encoder follows the same GOP table as in conventional HEVC, and uses the same CTU quadtree structure. Before encoding each block, the number of non-zero elements is counted: this is performed using the integral image of each foreground frame and then summing the elements in each block. If a block of content is detected which contains only zeros, then a novel prediction mode is triggered which we refer to as "background SKIP": the corresponding content from the decoded background is copied in the reconstruction, one bit is encoded in the bitstream for the entire block to signal the usage of background SKIP, and no other encoding is performed on the block. Conversely, if the block contains non-zero elements, HEVC encoding is performed as in conventional HEVC.

The process is illustrated in Figure 3.4.

Then the encoder considers then the next GOP and the following is performed:

1. The background reuse bit is read. If it is a 1, then the previous decoded background is copied to the current decoded background and nothing is encoded in the bitstream. Otherwise, the current background is encoded. The previous background is used as reference during the encoding.

2. The current GOP in the foreground sequence is then encoded as previously described.

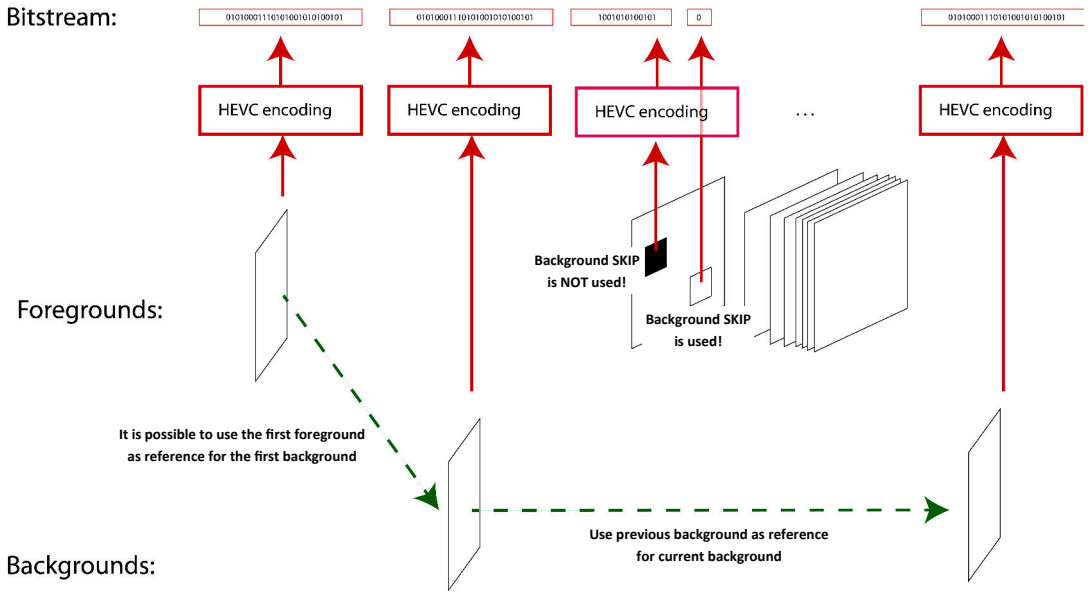


Figure 3.4: Adapted HEVC encoder for encoding Block-GOP LRSD output.

The encoding proceeds until all GOPs in the sequence are encoded.

3.3 Modified LRSD Model for HEVC

We need to adapt the low-rank and sparse decomposition in approximated RPCA to meet the specifications described in the previous section. In this section we will explain the modified LRSD model used for HEVC. We assume that each GOP is formed by p frames $\mathcal{F}_j, j = 1, \ldots, p$ of size $w \times h$. Also, we assume each frame consists of several blocks of size $b \times b$ (where w and h are multiples of b without loss of generality). If we had $w = h = 4$ and $b = 2$ the considered block structure of $\mathcal{F}_j$ would be

$$\mathcal{F}_j = \begin{bmatrix} f_{11} & f_{12} & f_{13} & f_{14} \\ f_{21} & f_{22} & f_{23} & f_{24} \\ f_{31} & f_{32} & f_{33} & f_{34} \\ f_{41} & f_{42} & f_{43} & f_{44} \end{bmatrix} = \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix} \tag{3.2}$$

So each frame in matrix A will be divided into such blocks, and since the data is arranged in A by concatenating each frame's pixels information by row-order in a column of A, the block structure would actually look like this

$$A_j = \begin{bmatrix} f_{11} & f_{12} & f_{13} & f_{14} & f_{21} & f_{22} & f_{23} & f_{24} & f_{31} & f_{32} & f_{33} & f_{34} & f_{41} & f_{42} & f_{43} & f_{44} \end{bmatrix}^T \tag{3.3}$$

The optimisation problem then is defined as

$$L, S = \underset{S, rank(L)=1}{\arg\min} \|A - L - S\|_F^2 + \lambda \|S\|_{\mathcal{B}}, \tag{3.4}$$

where the $\mathcal{B}$-norm of S is induced by the following block structure

$$S = \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1p} \\ s_{21} & s_{22} & \dots & s_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ s_{(\frac{w}{b^2})1} & s_{(\frac{w}{b^2})2} & \dots & s_{(\frac{wh}{b^2})p} \end{bmatrix}, \tag{3.5}$$

where the size of each block S_{ij} is $b^2 \times 1$. The $\mathcal{B}$-norm of S is defined as the sum of the ℓ_2-norm of its $b^2 \times 1$ column blocks S_{ij}, i.e.

$$\|S\|_{\mathcal{B}} \equiv \sum_{i=1}^{\frac{wh}{b^2}} \sum_{j=1}^{p} \|S_{ij}\|_2 \tag{3.6}$$

The $\mathcal{B}$-norm is an extension of the matrix norm that considers the column block structure of a matrix proposed in our previous work [40]. Hence, we essentially have

$$\|S\|_{\mathcal{B}} = \sum_{i,j} \|S_{ij}\|_2 = \|s_{11} \ s_{12} \ \dots \ s_{(\frac{wh}{b^2})p}\|_{2,1}, \tag{3.7}$$

where $\|\cdot\|_{2,1}$ is the $\ell_{2,1}$-norm and the $\mathcal{B}$-norm shares the same properties as the $\ell_{2,1}$-norm.

The solution to the optimisation problem (3.4) would be a compromise between the

reduction of the error in decomposition of A and the number of the non-zero $b \times b$ blocks in the foreground matrix S. The solution of the minimisation process would favour few non-zero blocks in S and then lots of zero blocks in between, which is the desired property for HEVC application. Following the technique we suggested in [40], we adopt an alternating strategy for solving the optimisation problem (3.4) with two separate subproblems

$$L^t = \underset{rank(L)\leq r}{\arg\min} \|A - L - S^{t-1}\|_F^2 \tag{3.8}$$

$$S^t = \underset{S}{\arg\min} \|A - L^t - S\|_F^2 + \lambda\|S\|_{\mathcal{B}} \tag{3.9}$$

The first subproblem is solved by updating L^t via singular value thresholding of the matrix $A - S^{t-1}$. For the second subproblem, a closed-form of the solution S^t is obtained using the following lemma. Assume $H = A - L^t$.

Lemma: Suppose the $m \times n$ matrix H has the following block structure:

$$H = \begin{bmatrix} h_{11} & \dots & h_{1p} \\ \vdots & \ddots & \vdots \\ h_{v1} & \dots & h_{vp} \end{bmatrix} \tag{3.10}$$

where each block h_{ij} is a $b^2 \times 1$ column vector. Then the matrix S that minimises the problem

$$\underset{S}{\arg\min} \|H - S\|_F^2 + \lambda\|S\|_{\mathcal{B}}, \tag{3.11}$$

is written as

$$S = \begin{bmatrix} s_{11} & \dots & s_{1p} \\ \vdots & \ddots & \vdots \\ s_{v1} & \dots & s_{vp} \end{bmatrix}, \tag{3.12}$$

where

$$S_{ij} = H_{ij} \max\left(0, 1 - \frac{\lambda}{\|H_{ij}\|_2}\right) \tag{3.13}$$

Proof: The objective function is expanded as:

$$\|H - S\|_F^2 + \lambda\|S\|_{\mathcal{B}} = \sum_{i,j} \|H_{ij} - S_{ij}\|_F + \lambda \sum_{i,j} \|S_{ij}\|_2 \tag{3.14}$$

The thesis of the lemma is obtained applying lemma 1, in the subproblems corresponding to each summand of the expression. ■

Since closed-form solutions of both subproblems exists, we can design a convergent algorithm where the objective function of the minimisation problem is decreasing. The obtained expression of the solution of the second subproblem means that in each iteration the algorithm is ignoring the column blocks of S where the ℓ_2-norm of the block is less than the parameter λ. It guarantees the desired block structure for the sparsity of the foreground frames.

3.3.1 LRSD model with a single background per GOP

The matrix L obtained as solution of the optimisation problem (3.4) is a rank-1 matrix by definition, i.e., L can be written as $L = [\alpha_1 l_1, \alpha_2 l_1, \ldots, \alpha_p l_1]$ where l_1 is the first column of L and α are scalars with $\alpha_1 = 1$. It means to reconstruct A, one needs a $w \times h$ vector l and the p coefficients α. Then these coefficients must be encoded as an array of numbers by HEVC. In order to get a rank-1 matrix L with identical columns, we propose the following modification of the model (3.4)

$$L, S = \arg\min_{l,S} \|A - l \times \mathbb{1} - S\|_F^2 + \lambda\|S\|_{\mathcal{B}}, \tag{3.15}$$

where $\mathbb{1} = [1, 1, \ldots, 1]$ and with the same length as l. Following the same alternating strategy, the subproblem to be solved now in each iteration is

$$l^t = \arg\min_{l} \|A - l \times \mathbb{1} - S^{t-1}\|_F^2 \quad \text{where} \quad L^t = l^t \times \mathbb{1} \tag{3.16}$$

In other words, we need to find the matrix with identical columns that is closest to a given matrix. But, we know that the rank-1 matrix that is closer to a given matrix is unique and it can be written as $\sigma_1 U_1 V_1^T$, where σ_1, U_1, and V_1 are the largest singular value, left singular vector, and right singular vector of the matrix respectively. Then, the solution vector l of (3.16) is given by the minimisation problem below

$$l^t = \arg\min_{l} \|\sigma_1 U_1 V_1^T - l \times \mathbb{1}\|_F^2 \tag{3.17}$$

It is easy to prove that the vector of the form $\sigma_1 V_{1j} U_1$ – where V_{1j} is the coefficient of the largest right singular vector of the matrix – minimises this expression.

3.3.2 LRSD-HEVC for sequences with moving camera

HEVC is not very efficient for videos that are captured with camera motion. LRSD can be performed with such video sequences, as one could assume that the camera motion exhibits itself as the global background motion, and therefore, all the pixels in the background obey this motion.

Here, we propose a modification of the model (3.15) including new variables to describe the global background motion. We further extend our work in [40] and [39] where an approximated RPCA model is extended to handle camera-induced motion. So for each GOP we would have

$$L, S, \tau = \arg\min_{l,S,\tau} \|A \circ \tau - l \times \mathbb{1} - S\|_F^2 + \lambda \|S\|_{\mathcal{B}}, \tag{3.18}$$

where τ is a vector describing the 2D affine, or 2D projective transformation parameters. The solution of the optimisation problem is again obtained using an alternating strategy as follows:

$$\tau^t = \arg\min_{\tau} \|A \circ \tau - l^{t-1} \times \mathbb{1} - S^{t-1}\|_F^2 \tag{3.19}$$

$$L^t = \arg\min_{l} \|A \circ \tau^t - l \times \mathbb{1} - S^{t-1}\|_F^2 \tag{3.20}$$

$$S^t = \arg\min_{S} \|A \circ \tau^t - l^t \times \mathbb{1} - S\|_F^2 + \lambda \|S\|_{\mathcal{B}} \tag{3.21}$$

The solution of the first and second subproblems are described in [40] and [39]; and the solution of the third subproblem is the same as we explained before in the lemma of section 3.3.

3.3.3 Low-bitrate background generation

As discussed before, the presented LRSD-HEVC framework will need to encode a foreground sequence that contains the sparse component, a background sequence per GOP, and a background reuse bit sequence to determine whether the previous GOP's background could be reused for the current GOP. To achieve lower bitrate and higher compression we propose a novel technique where we omit encoding parts of the background that are overlaid by the corresponding foreground pixels in the respective GOP.

As a simple example imagine a person moving across the scene in a GOP. There are CU's in the sparse matrix that for that GOP are always non-zero – this is almost always the case with a GOP size of 8 frames, since foreground does not move very fast in such a short period of time. Therefore, at the decoder side, when reconstructing the sequence, the information in CUs of background that correspond to the locations of always non-zero CU's in the foreground, is never used. Therefore, we must detect these CU locations and encode them as zero blocks.

From the coding perspective, the idea is that we do not need to re-encode the entire background because some portions may not be needed. A portion is not needed if there is no foreground frame for which that portion, under the correct transformation, is used in such foreground frame. So basically, we are looking for those pixels in the background such that, when they are transformed, they are never actually used in any of the foregrounds in a GOP. If a block of 8×8 pixels in the un-transformed background

contains only such pixels, then we do not need to encode that block.

The steps required for low-bitrate background generation are:

1. We take a given foreground at a certain time index, and we identify the blocks that are actually part of the foreground, then we transform them to the "coordinates" of the background;

2. We identify those pixels as "unnecessary", as they are "covered" by foreground;

3. We do the same for all foreground frames. We then identify pixels that were marked as "unnecessary" in all foregrounds. Then those pixels are truly "unnecessary". If a block of 8×8 pixels in the background contains only unnecessary pixels, we assign that whole block to zero.

An example of the low-bitrate background generation is shown in Figure 3.5.

3.3.4 Variable GOP size

In section 3.3.1 we introduced a mechanism that adapts LRSD to create a single background per GOP. Sometimes, it might be possible to reuse the same background for several consecutive GOPs Based on this idea the size of each GOP is determined by an adaptive algorithm that minimises the number of non-zero blocks in the background and foregrounds. The GOP size would always be a multiple of 8.

We start by performing the LRSD for the first GOP. Then we calculate the average number of non-zero blocks in the GOP, that would give us an estimation of how well the calculated background can cover most of the information for reconstruction of that GOP; i.e., if a good background model is obtained, the foreground matrix will contain many more zero blocks. Then we proceed to the second GOP, and calculate a background model for the second GOP. Then we calculate two foreground matrices, the first of which uses the second GOP background, and the second of which uses the first GOP background.

Then the average number of non-zero blocks for each of these foreground matrices is calculated. If the background of the first GOP could still be a good model for the second GOP we expect that the number of non-zero blocks in this case be less than the number of non-zero blocks of the foreground calculated with second GOP background. Then we discard the second background and increase the GOP size this time to 16 (first GOP and second GOP are merged). The same is repeated for the third GOP and so on. We allow a maximum GOP size of 64 in this case. We then obtain a sequence of GOP size numbers that need to be encoded separately. The advantage this method gives is saving on bitrates when encoding background parts.

This idea can be extended to cases where the sequence has been taken with a moving camera. Then in this case, instead of the non-zero blocks, the transformation parameters that describe the background displacement will determine the size of each GOP. Usually when calculating the background motion parameters we expect that the horizontal and vertical displacement vectors from one GOP to the next do not exceed 10% of the size of the image. For example for a full-HD image with dimensions 1920×1080 pixels we allow a maximum 192 pixels displacement in horizontal axis and 108 pixels displacement in vertical axis. If then the displacement for the current GOP relative to the previous GOP falls under these limitations, we merge those two GOPs and we use the same background for both. Again, this is performed for consecutive GOPs with a maximum GOP size of 64 allowed.

3.4 Model Analysis

We have conducted qualitative and quantitative tests for the efficiency of the proposed LRSD-HEVC method. The results reported here are all the output of the LRSD-HEVC algorithm before being encoded and decoded by standard HEVC, as the encoding and decoding processes are out of the scope of this thesis.

We present some quantitative results, each testing one aspect of the proposed modi-

fications to LRSD for HEVC. We have two main sets of results for each category:

- Core model, which is an incremental version with an unchanging GOP size, and,
- Variable GOP model, which is a modification of the core model with the adaptive GOP size selection method described in section 3.3.4.

From here on we refer to the core model as "core" and the variable GOP model as "vGOP". Our tests have been conducted for 7 standard HEVC test sequences with 64 frames, down-sampled from UHD quality (3840×2160 pixels) to full-HD (1920×1080 pixels). For all the tests the results are obtained with a set of 20 tuning parameters λ linearly distributed in the range $[0, 0.25]$.

3.4.1 Single background per GOP vs. multiple backgrounds per GOP

We have tested how using a single background per GOP (first frame background L_1) can affect the overall reconstruction quality of each sequence in terms of Peak Signal to Noise Ratio (PSNR) and the number of non-zero blocks. Possibly we could understand "bitrate" better than the "number of non-zero blocks". But a bitrate measure can only be achieved once the sequence is encoded and decoded back with standard HEVC, and this falls out of the scope of this thesis. However, we expect that the two measures be closely correlated, and hence, we use these terms interchangeably.

Figure 3.6-(left) shows the performance of the core model using a single background per GOP vs. a background per frame in the GOP. Fixed parameters are: GOP size $p = 8$, CTU quadtree division enabled $QTD = 1$, maximum CU size $blocksize = 8$. It can be seen that as the number of non-zero blocks increases, the model with multiple backgrounds per GOP that are calculated by the SVD, more or less obtains the same PSNR as the non-SVD version. However, one could argue that to obtain the same PSNR we would need to use higher bitrate to encode the additional non-zero blocks required by the non-SVD model. Nevertheless, the margins between the two models are mostly very

small and negligible. The same can be observed for the vGOP model. Figure 3.6-(right) shows the performance of vGOP using a single background per GOP vs. a background per frame in the GOP. Fixed parameters are: GOP size $p = variable$, CTU quadtree structure enabled $QTD = 1$, maximum CU size $blocksize = 8$.

3.4.2 CTU quadtree division

In this test we demonstrate the effect the CTU quadtree division has on the PSNR vs. number of non-zero blocks. The first test was performed with CTU quadtree division enabled ($QTD = 1$) and the second without the division ($QTD = 0$), where the maximum and minimum blocksize (CU size) is the same.

Figure 3.7-(left) shows the performance of core model. Fixed parameters are: GOP size $p = 8$, maximum CU size $blocksize = 8$. It can be observed that from the model's standpoint for a given λ, when refinement is used ($QTD = 1$), PSNR is lower while less non-zero blocks are used. Conversely, disabling the refinement ($QTD = 0$) increases PSNR for the price of higher bitrate. However, the other way to interpret these results is that for a given bitrate, using the refinement ($QTD = 1$) gives higher PSNR than that of disabling the refinement ($QTD = 0$).

Figure 3.7-(right) shows the performance of vGOP. Fixed parameters are: GOP size $p = variable$, maximum CU size $blocksize = 8$. For the vGOP model however, refinement does not affect the quality or bitrate. This is due to the fact that the vGOP already takes into account the trade-off between bitrate and quality.

3.4.3 CU size

To find the optimal CU size (blocksize) we have conducted the following tests with similar parameter settings as before. We have tested two maximal CU sizes 8 and 16. Figure 3.8-(left) shows the performance of core model, and Figure 3.8-(right) shows the

performance of the vGOP model. In both models for a given λ, a higher CU size favours the quality for the price of higher non-zero blocks, and thus higher bitrate. The other way to interpret these results is that for a given bitrate using a CU size of 16 gives better PSNR on average – although this interpretation does not always hold, specifically for the test sequences BQMall, HoneyBee, ShakeNDry, and TrafficFlow in the vGOP model.

3.4.4 GOP size

We have tested the same sequences in the same parameter settings with different GOP sizes that are all multiples of 8, i.e., $p = \{8, 16, 24, 32, 40, 48, 56, 64\}$. For the vGOP model p will determine the starting GOP size, and can grow if vGOP determines so. Figures 3.9 shows the effect of GOP size for each sequence in core model, for 20 different λ values. Interestingly, for all examples a lower GOP size is almost always favourable for both quality and bitrate. This means that generation of a background that can describe most of the pixel content in each GOP is of utmost importance. As the GOP size is increased, this background model becomes less and less descriptive of the unchanging pixel content of the GOP, which is to be expected. It is evident that using a background for less number of frames, decreases the bitrate in the foregrounds. However, this entails that more background images would need to be encoded by the HEVC. The effect this could have on HEVC is interesting and needs to be further studied. Figure 3.10 shows the effect of GOP size for each sequence in the vGOP model.

In Table 3-A quantitative results for the four test sequences have been shown. For each sequence we varied λ that controls the quality of reconstruction in the core model, maximum CU size *blocksize*, and GOP size. We enabled CTU quadtree division for all the tests. Then we obtain PSNR values of the reconstruction quality using the background and foreground sequences. As mentioned, we report these results before encoding and decoding by HEVC takes place. Two PSNR values are reported here; the mean PSNR for each sequence, and the maximum PSNR achieved for a given frame of each sequence. These results indicate that generally we obtain better reconstruction

quality using a combination of the smallest possible non-zero λ, a CU size $blocksize = 16$, and a GOP size $p = 8$.

3.5 Summary

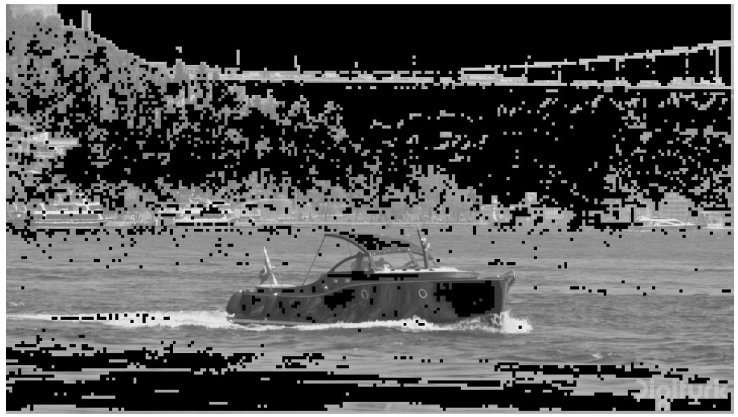
In this Chapter we described a new low-rank and sparse decomposition method for HEVC applications, and conducted tests to evaluate every aspect of the proposed methodology. In future work, we would investigate further how our model will affect quality vs. bitrate after decoding and encoding processes are performed by HEVC. Optimal parameter selection has been an area of interest in the proposed method, and as such we have left enough room for reconfigurability of our model to achieve the best trade-off when used in conjunction with an HEVC encoder/decoder.

Table 3-A: LRSD-HEVC quantitative results for four full-HD sequences.

Sequence	λ	block size	GOP size	mean PSNR	max PSNR	Sequence	λ	block size	GOP size	mean PSNR	max PSNR
BasketballDrive	0.005	8	8	45.24	56.45	BQTerrace	0.005	8	8	∞	∞
			16	44.81	48.99				16	70.20	89.17
		16	8	45.23	56.29			16	8	∞	∞
			16	44.77	48.55				16	70.24	89.68
	0.01	8	8	39.98	42.02		0.01	8	8	55.81	63.58
			16	39.26	41.24				16	52.50	61.36
		16	8	39.97	42.01			16	8	55.98	63.48
			16	39.26	41.25				16	51.92	59.70
	0.015	8	8	37.90	40.19		0.015	8	8	49.18	53.60
			16	36.63	39.07				16	46.72	52.10
		16	8	37.91	40.26			16	8	49.20	53.63
			16	36.65	39.08				16	46.73	52.01
	0.02	8	8	36.34	38.92		0.02	8	8	45.78	50.50
			16	34.81	37.51				16	43.39	48.69
		16	8	36.35	38.91			16	8	45.74	50.43
			16	34.75	37.52				16	46.28	49.14
Bosphorus	0.005	8	8	73.30	93.44	ChristmasTree	0.005	8	8	56.74	93.37
			16	61.33	74.76				16	52.14	81.74
		16	8	73.77	98.74			16	8	56.64	93.37
			16	61.06	74.61				16	52.05	82.97
	0.01	8	8	54.52	59.03		0.01	8	8	41.48	52.00
			16	50.97	55.97				16	40.40	47.51
		16	8	54.30	59.70			16	8	41.49	52.16
			16	50.76	55.12				16	40.41	47.59
	0.015	8	8	49.62	54.17		0.015	8	8	38.46	42.94
			16	46.93	50.76				16	37.33	41.53
		16	8	49.38	53.25			16	8	38.46	42.97
			16	46.95	50.46				16	37.32	41.56
	0.02	8	8	47.07	50.99		0.02	8	8	36.59	40.58
			16	44.70	47.47				16	35.21	39.47
		16	8	46.98	49.17			16	8	36.60	40.57
			16	44.78	48.07				16	35.20	39.59

(a) Background with low-bitrate

(b) Actual background



(c) Sparse foreground

(d) Reconstruction with (a) and (c) with PSNR = 59.86

Figure 3.5: Low-bitrate background generation and corresponding reconstruction results. The low-bitrate background in (a) is generated from the actual background in (b). Then (a) and (c) are used to reconstruct the original frame in (d).

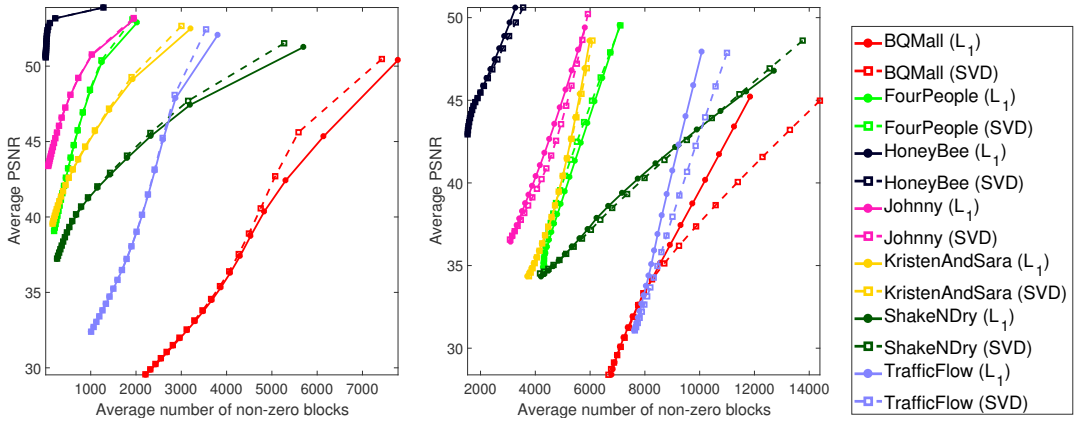


Figure 3.6: PSNR vs. non-zero blocks for single background per GOP (L1 solid lines) and multiple backgrounds per GOP (SVD dashed lines). Left: core model. Right: vGOP model.

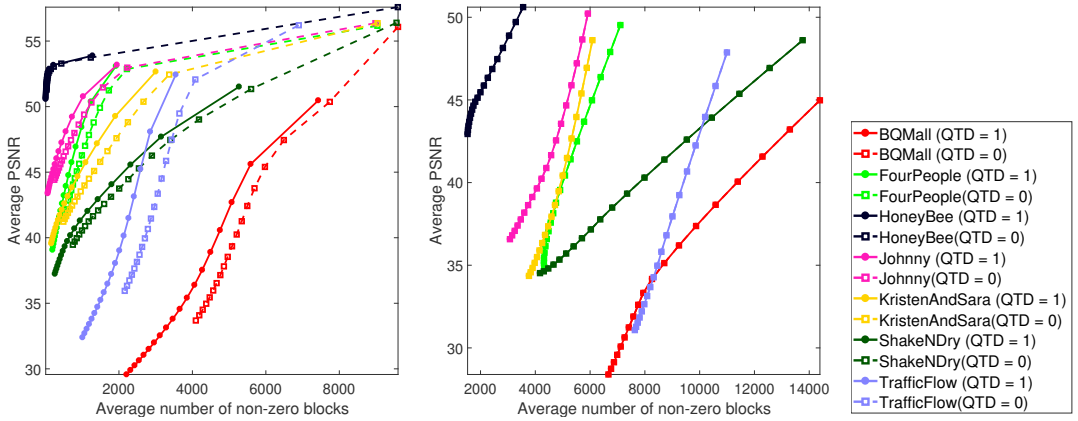


Figure 3.7: PSNR vs. non-zero blocks with QTD (solid lines) and without QTD (dashed lines). Left: core model. Right: vGOP model.

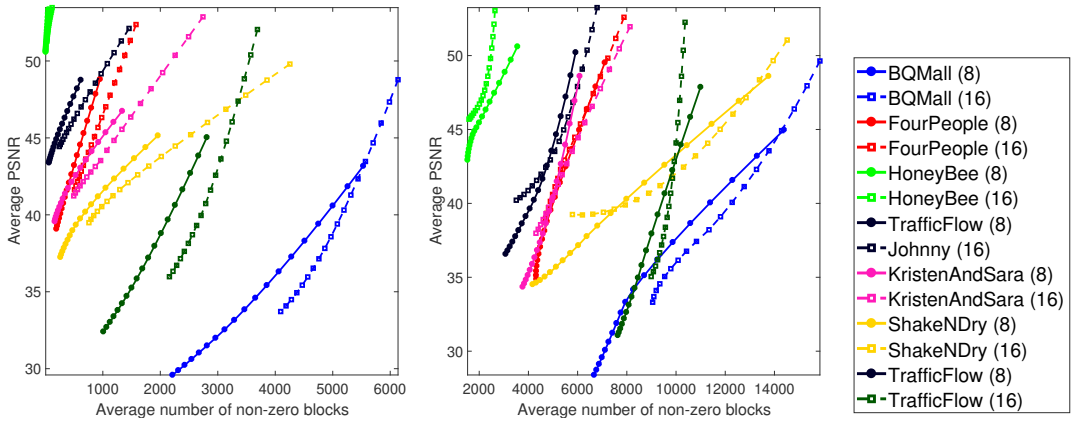


Figure 3.8: PSNR vs. non-zero blocks with CU size 8 (solid lines) and CU size 16 (dashed lines). Left: core model. Right: vGOP model.

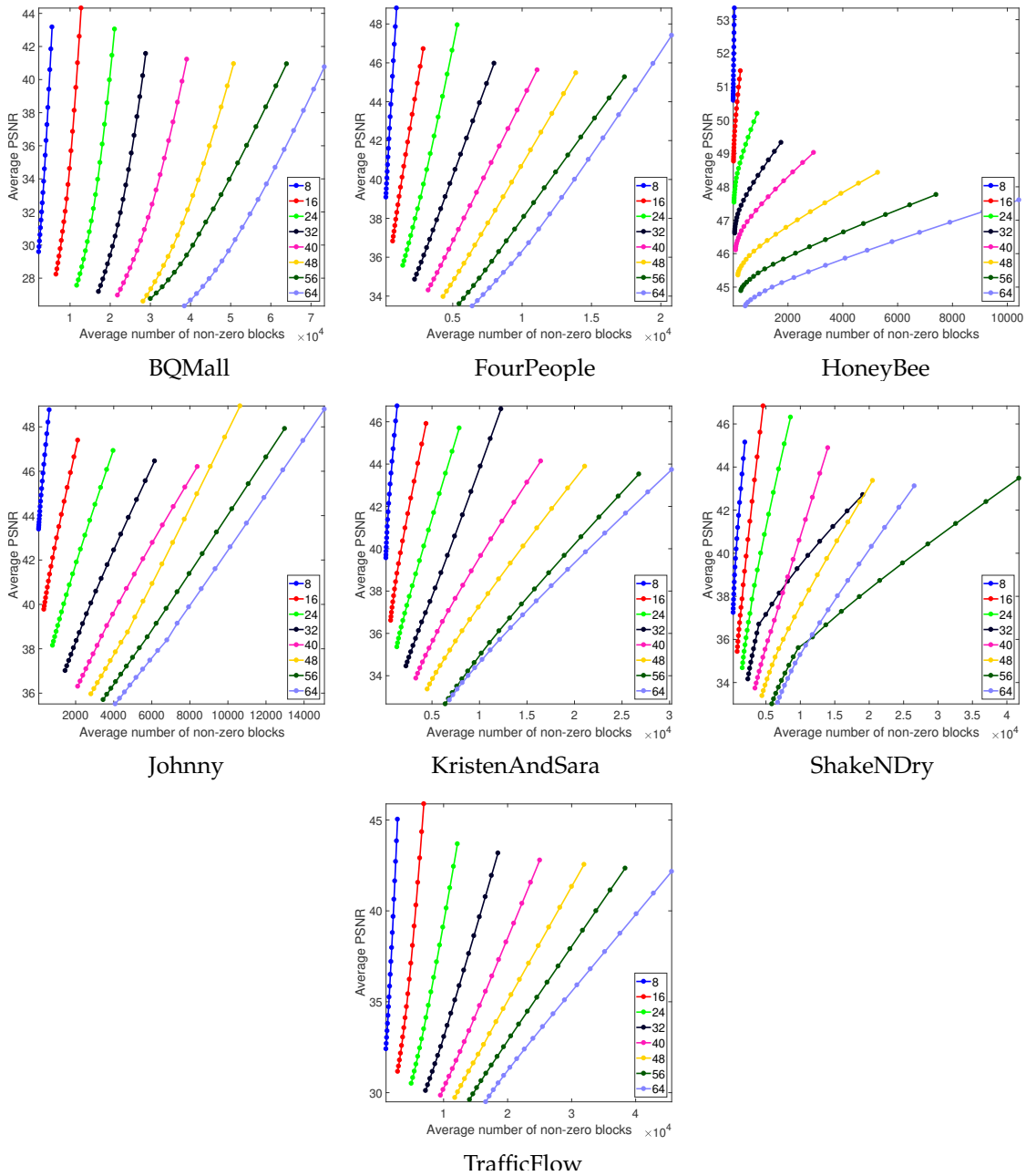


Figure 3.9: PSNR vs. number of non-zero blocks for the core model for various GOP sizes for 7 test sequences

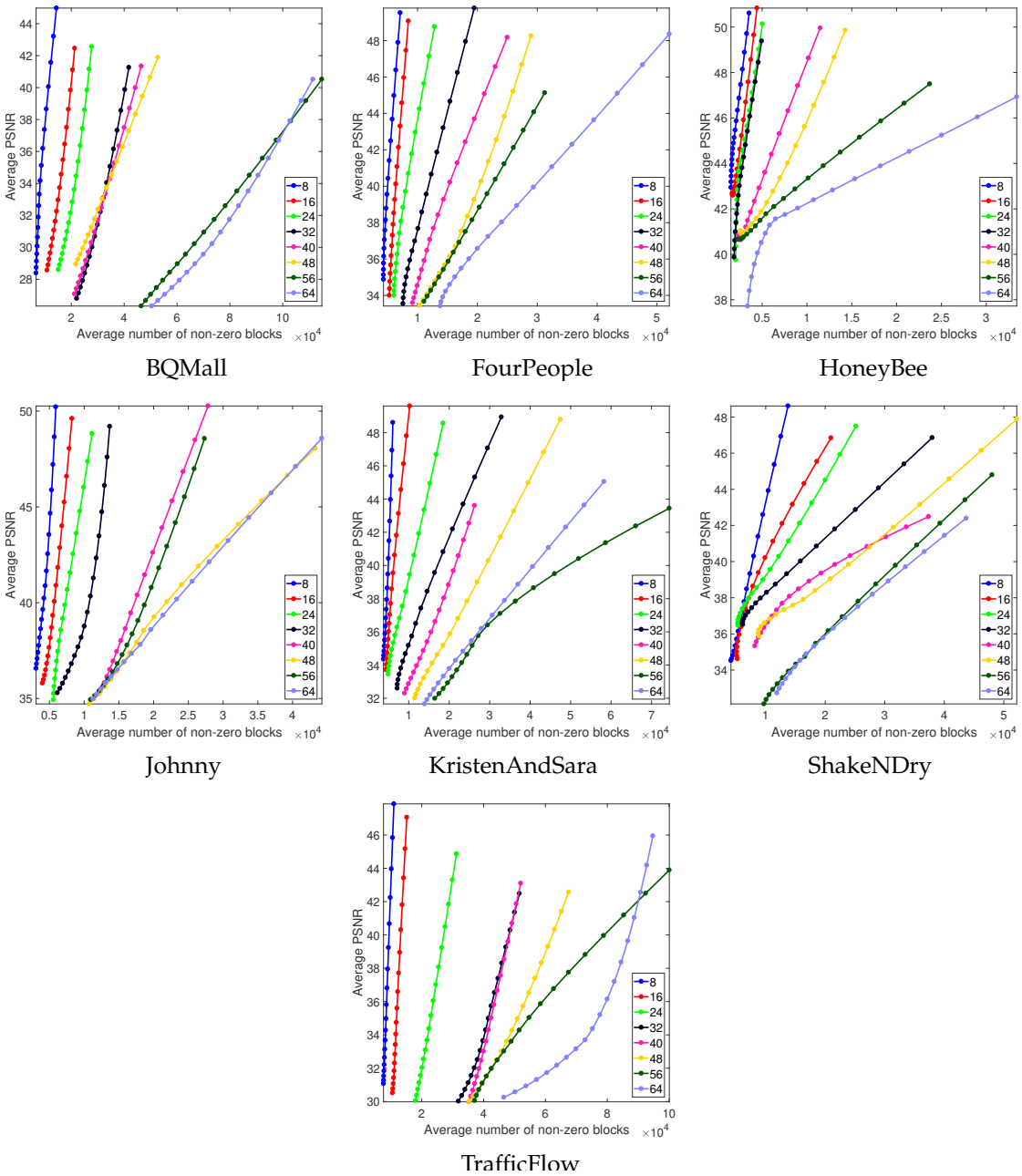


Figure 3.10: PSNR vs. number of non-zero blocks for the vGOP model for various GOP sizes for 7 test sequences

Chapter 4

Alignment and Recovery of Corrupted and Linearly Correlated Images and Video Frames

In this chapter we present an approximated Robust Principal Component Analysis (ARPCA) framework for recovery of a set of linearly correlated images. Our algorithm seeks an optimal solution for decomposing a batch of realistic unaligned and corrupted images as the sum of a low-rank and a sparse corruption matrix, while simultaneously aligning the images according to the optimal image transformations. This extremely challenging optimisation problem has been reduced to solving a number of subproblems, that minimise the sum of Frobenius norm and the ℓ_1-norm of the mentioned matrices, with guaranteed faster convergence than the state-of-the-art RPCA-based algorithms. The efficacy of the proposed method is verified with extensive experiments with real and synthetic data. The findings of this chapter are published in [39].

4.1 Introduction

In recent years, the popularity of image and video sharing websites such as Facebook, Instagram, YouTube, etc. has led to a dramatically large amount of data becoming available online. Applications such as face, digit, and object recognition is a problem domain in computer vision where low-dimensional linear models have received a great deal of attention. The available substantial data can be very challenging (if not impossible) to process with computer vision algorithms, if the difficulties such as significant illumination variation, occlusion, misalignment, deformities, and noise are not dealt with using a proper method. The most challenging task is aligning a set of images of an object to a fixed canonical template, simultaneously with removing occlusions, corruptions, and specularities to obtain an accurate representation of the object of interest based on similarity, for robust recognition or classification.

A great deal of progress has been made in batch image alignment, the most notable of which is [124], where the authors used a similar convex relaxation program in which the transformed images of an object from a set of unaligned images were decomposed as the sum of images from a low-rank approximation, and sparse large errors. Their algorithm was successful in cases of rigid and parametric classes of transformations, given the amount of misalignment and corruption was within a limited bound, and image sizes were not too big. While their method demonstrated robustness to corruption and occlusion, it uses a very expensive optimisation program, based on a Lagrangian multipliers iterative linearisation, whose performance is slow in applications where real-time or very fast performance is sought. Another work [154], minimises a rank surrogate, however lacks robustness to corruption and occlusion. In addition, the canonical frame that their algorithm could handle was a small image of 49×49 pixels with only a small Euclidean transformation and limited corruption.

In this chapter, a new algorithm is introduced for recovery of linearly correlated images and video frames, despite occlusions, corruptions, and large misalignment. Our

method builds on recent advances in rank minimisation and formulates the problem of batch image alignment as the solution of a subproblem in the sequence of subproblems. The solution of these problems have been shown to be efficient in our preceding work [40], [39]. Our algorithm can handle batches of high resolution (up to HD quality) images in several minutes. We verify the efficacy and accuracy of our algorithm as well as its superiority to similar methods, with extensive experiments on unconstrained real images with wide range of corruption and misalignment. These results suggest the potential of our algorithm as a general tool for video stabilisation, compression, and object tracking.

4.2 Approximated RPCA Framework

Suppose we are given n unaligned images or video frames $I_1, \ldots, I_n \in \mathbb{R}^{w \times h}$ of an object. We produce a matrix $A = [A_1, \ldots, A_n] \in \mathbb{R}^{m \times n}$ by concatenating all elements of I in row-order as columns of A. The matrix A should then be *low-rank* – since its columns are linearly correlated – with a low-rank component L, and the large errors can be expressed as the sum of a sparse matrix S and a noise matrix G, while the parametric transformations τ can model the potential global misalignment.

$$A \circ \tau = L + S + G \tag{4.1}$$

$A_j \circ \tau_j$ denotes the j-th frame after transformation parameterised by the vector $\tau_j \in \mathbb{R}^\rho$ where ρ is the number of parameters fully describing the global motion model. Therefore $\rho = 4$ corresponds to similarity, $\rho = 6$ to affine, and $\rho = 8$ to projective transformation. It was shown that for the problem of recovering low-rank matrices from sparse errors, as long as the rank of the matrix A to be recovered is not too high and the number of the errors is not too large, minimising the natural convex surrogate for $rank(A) + \lambda \|S\|_0$ (with λ soft-thresholding parameter) can *exactly* recover A [17]. Here, we use a different relaxation that replaces $rank(\cdot)$ with the *Frobenius norm*: $\|A\|_F = \sqrt{\sum_{i,j} A_{ij}^2}$, and the ℓ_0-norm $\|S\|_0$ with the ℓ_1-norm: $\|S\|_1 = \sum_{i,j} |S_{ij}|$ in an approximated noisy case. Applying

this relaxation to (4.1) yields a new optimisation problem, such that $A \circ \tau \approx L + S$:

$$\underset{\substack{L,S,\tau \\ rank(L)\leq k}}{\arg\min} \ \|A \circ \tau - L - S\|_F^2 + \lambda \|S\|_1 \tag{4.2}$$

The authors in [6] showed that for convex, Lambertian objects, images taken under distant illumination lie near an approximately nine-dimensional linear subspace known as the *harmonic plane*. However, with face images which are neither perfectly convex nor Lambertian, this low-rank model is violated, due to cast shadows, specularities, occlusions, and misalignment. These errors are large in magnitude, but often sparse in the spatial domain. Given a sufficient number $n > rank(A)$ of those images, the extremely efficient and computationally inexpensive approximated Robust Principal Component Analysis in (4.2) will be able to remove those errors, as well as align all those images in the same canonical template. To solve this problem we use an alternating strategy minimising the function for three parameters L, S, and τ one at a time until convergence; for a fixed λ the iterative process below will have a monotonically decreasing value, converging to a local minimum:

$$\tau^t = \underset{\tau}{\arg\min} \|A \circ \tau - L^{t-1} - S^{t-1}\|_F^2 \tag{4.3}$$

$$L^t = \underset{rank(L)\leq k}{\arg\min} \ \|A \circ \tau^t - L - S^{t-1}\|_F^2 \tag{4.4}$$

$$S^t = \underset{S}{\arg\min} \|A \circ \tau^t - L^t - S\|_F^2 + \lambda \|S\|_1 \tag{4.5}$$

The main remaining difficulty in solving (4.2) is the non-linearity of the constraint $A \circ \tau \approx L + S$, which arises as a result of the dependence of $A \circ \tau$ on the transformations τ. We use the linearisation method described in [40], where an incremental refinement is used. The i-th geometric transformation is comprised of a parameter vector τ_i, $i = 1, \ldots, n$ where different spatial transformations can be considered. We use the $2D$ parametric transforms to model the translation, rotation, and planar deformation in

the low-rank subspace. We obtain an initial approximation for the parameters τ_i using a feature matching, indirect method with SIFT features [104] where the images are aligned to the middle image. This method is more robust and much faster compared to direct methods used in [124] with larger image sizes and more extreme parametric transformations and large camera parallax, displacement, and motion blur. Finally, in (4.3) we use the multi-resolution incremental refinement described in [147], to estimate these motion parameters. To calculate the rank-k matrix that is the nearest estimate of the matrix $A \circ \tau^t - S^{t-1}$ in (4.4), SVD gives a closed-form solution as: $L^t = \sum_{i=1}^{k} \sigma_i U_i V_i^T$, with the coefficients σ_i the singular values, and the vectors U_i and V_i the singular vectors of the matrix $A \circ \tau^t - S^{t-1}$. Finally in (4.5) the matrix S^t is updated using the parameter λ acting as a regulating parameter, where the elements of the matrix $A \circ \tau^t \leq \lambda$ are considered zero.

4.3 Experiments

In this section, we demonstrate the efficacy of our method in a variety of image recovery tasks. We verify the correctness of our method with controlled and uncontrolled examples, and show that it outperforms state-of-the-art methods in recovery of corrupted data while simultaneously compensating for any misalignment. Our realistic examples are taken from the challenging Labelled Faces in the Wild (LFW) database [76]. Experiments on video data and handwritten digits further indicate the generality of our method for various applications. Moreover, our algorithm can handle more complicated deformations and transformations such as planar homographies as shown in one of the tests, which indicates wide range of applications in video stabilisation and compression.

4.3.1 Speed of our method

For this example, on a 3.40GHz (single core) Intel Core i7-4770 machine with 32GB of RAM our MATLAB implementation requires 11.07 seconds to recover and align 100

perturbed and corrupted synthetic images of size 49×49, whereas [124] requires 41.44 seconds. Moreover, our algorithm is able to handle large image sizes (up to HD quality), which demonstrates impressive computational efficiency as a direct result of using our approximated RPCA optimisation framework.

4.3.2 Removing shadows and specularities from face images

We test our algorithm using a set of controlled images. Figure 4.1 shows 100 images of a dummy head that are perturbed and occluded randomly. The images are all 49×49 pixels (our algorithm can handle much larger image sizes, however for comparison with similar methods the same image data have been used). To each image a random Euclidean transform is applied with angle of rotation uniformly distributed in $[-10\,^\circ, 10\,^\circ]$ and x- and y-translations are uniformly distributed in $[-3, 3]$ pixels, while 6% of the pixels are corrupted. Notice that our method correctly removes the occlusions (Figure 4.1-(d)), to produce a rank 3 matrix of well-aligned images (Figure 4.1-(c)). RASL [124] can produce the same results but with the unnecessarily high minimised rank 48. The rank 3 matrix best describes the general appearance of the face image in this case, while preserving the prominent features for recognition purposes.

Next, we validate our approach using more challenging images taken from Labelled Faces in the Wild (LFW) [76] dataset of public figures. These images exhibit significant variations in pose and facial expression, illumination, and occlusion; moreover, the ground truth (i.e. undistorted, not rotated, not shifted) image is not known. The images are aligned to a 80×60 canonical frame, and affine transformations are used to cope with large variability in poses. Figure 4.2 shows one example from this dataset. Notice the average face after alignment is significantly clearer in Figure 4.2-(f) indicating improved alignment achieved by our method. This example demonstrates our method's ability in correcting errors in real images, which could be used to improve the performance of current face recognition systems. Figure 4.3 shows another example before and after alignment from the LFW dataset. In this example it can be seen that in 4.3-(c) the

aligned and corrected image is recovered from the set of corrupted images, and can be used readily for face recognition applications. Figure 4.4 shows more examples from the LFW dataset, before and after alignment and recovery. In this example 35 images were used per subject to obtain the results.

4.3.3 Recovery of corrupted and misaligned handwritten digits

Our method can be applied to aligning any general set of images with strong linear correlation. In this test, we used 100 handwritten digits "3" from the MNIST database in Figure 4.5. Our algorithm can obtain comparably good performance on this example despite the fact that it does not explicitly target binary image alignment.

4.3.4 Recovery of deformed and corrupted planar surfaces

In this example, our algorithm is applied to images that differ by planar homographies, to demonstrate how it can be used with more complicated deformation models. Figure 4.6 shows 8 images of a building, taken from various viewpoints by a perspective camera. As seen here, the algorithm correctly aligns the windows and removes branches occluding them. This hints useful applications for our method in image matching, mosaicking, and inpainting.

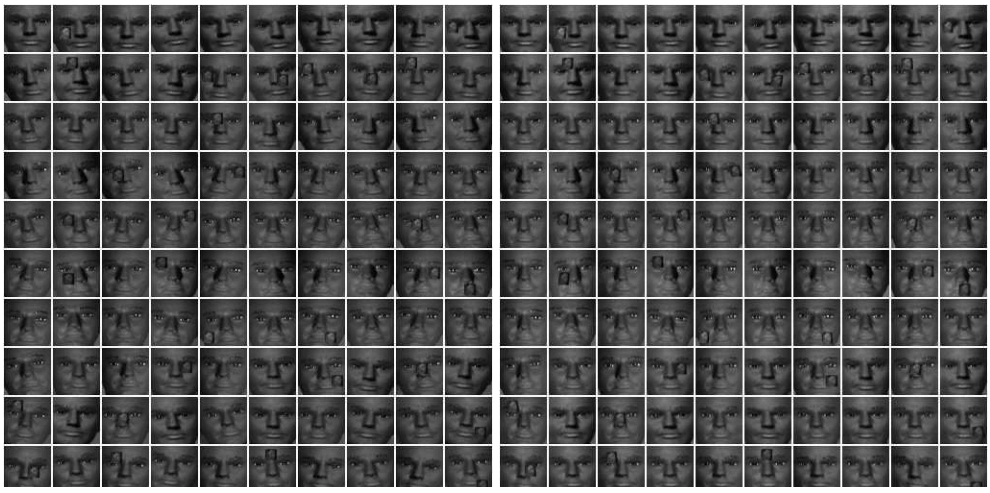
4.3.5 Video stabilisation for recovery of object of interest

Video frames taken from the same scene are usually linearly correlated. In this test, we demonstrate the ability of our method in aligning frames taken from a video. Figure 4.7 shows frames from a 140 frame video of Al Gore talking, obtained by applying a face detector to each frame individually. Due to imprecision in face detector there is high jitter from frame to frame. Next, we use affine transformations to obtain a well-aligned set of frames, and then we demonstrate a low-rank approximation of the frames as well

as the removed shadows, occlusions, and errors from the images. Notice that the errors shown in Figure 4.7-(d) compensate for local motion such as mouth movements, and eye blinking which are not considered in the global motion model. For this video with resolution 80×60 pixels with 140 frames our method needs 9.79 seconds while [124] takes 57.52 seconds to produce visually similar results. These results suggest the potential of our algorithm as a general tool for video stabilisation, compression, and object tracking.

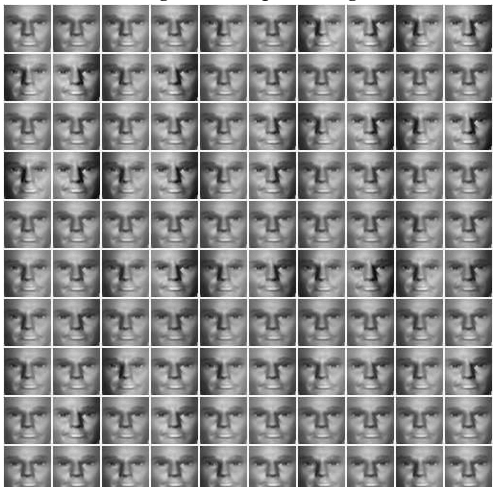
4.4 Conclusion

In this chapter we demonstrated the surprising effectiveness and efficacy of our approximated RPCA method for batch image recovery from corruptions and misalignment, and suggested applications such as batch image alignment, recovery of face images form corrupted data for face recognition, video stabilisation, image mosaicking, and image inpainting etc. Our proposed formulation directly impacts the speed of convergence of the algorithm, making it suited for real-time applications. One of the most important questions for future work is how to extend our framework to more general classes of transformations such as non-rigid and non-parametric that are exhibited in general video data, while providing the same practical guarantees for the amount of misalignment and corruption it can handle.

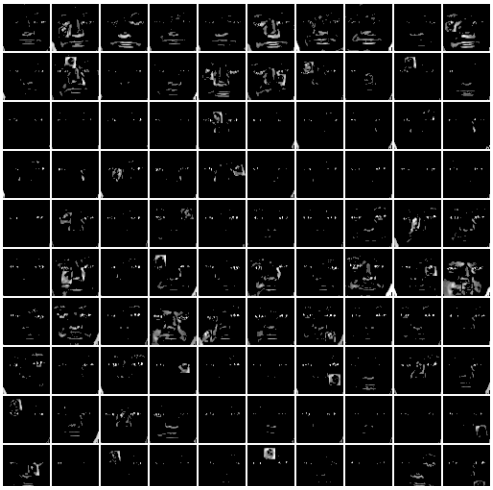


(a) Original corrupted images A

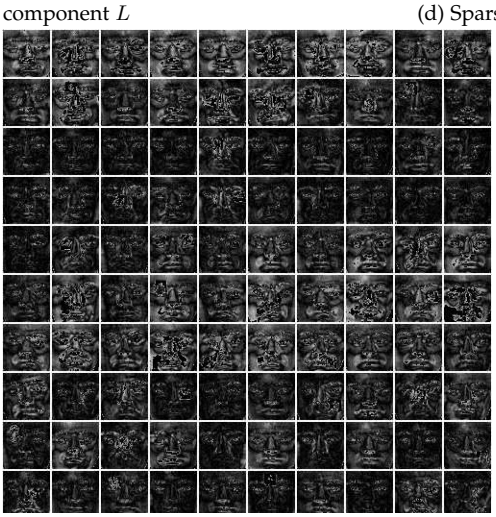
(b) Aligned images $A \circ \tau$



(c) Low-rank component L



(d) Sparse part S



(e) Errors G

Figure 4.1: Robust alignment by sparse and low-rank decomposition in Synthetic face images.

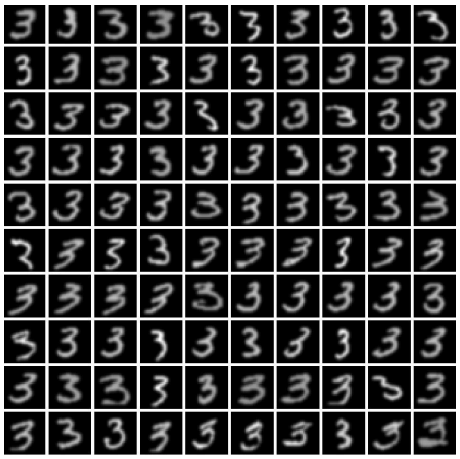
Figure 4.2: Robust alignment by sparse and low-rank decomposition in LFW dataset [76]. Contrast has been normalised in (d) and (e) for better visualisation. Figures (f), (g), (h), (i), and (j) correspond to the average of (a), (b), (c), (d), and (e) respectively.

(a) (b) (c) (d)

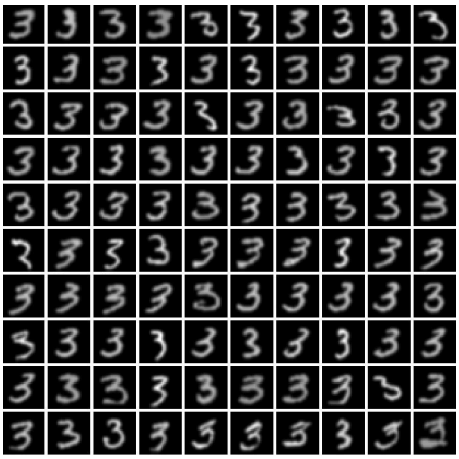
Figure 4.3: Removing shadows and corruptions on faces from LFW dataset [76]. (a), (b), (c), and (d) correspond to average of: original images, aligned images, low-rank component, and sparse specularities respectively.

(a) (b)

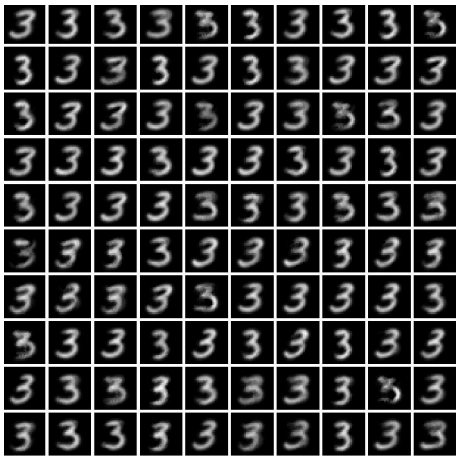
Figure 4.4: Face alignment in large datasets. Average faces (a) before and (b) after alignment and removal of shadows, corruptions, and specularities in LFW dataset [76] for 35 images per subject.



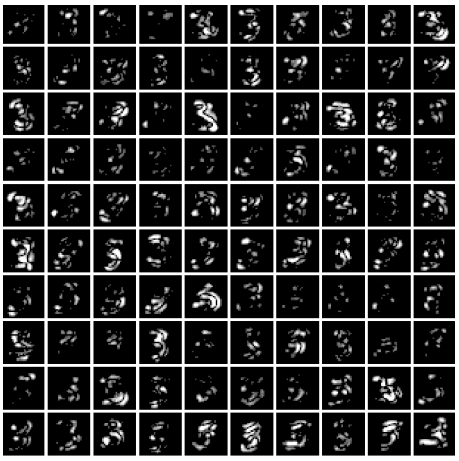
(a) Original handwritten digits A



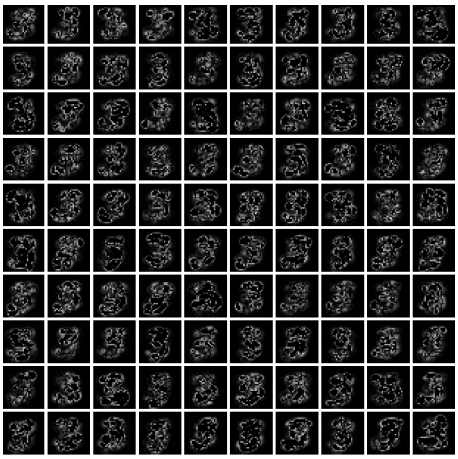
(b) Aligned digits $A \circ \tau$



(c) Low-rank component L



(d) Sparse part S

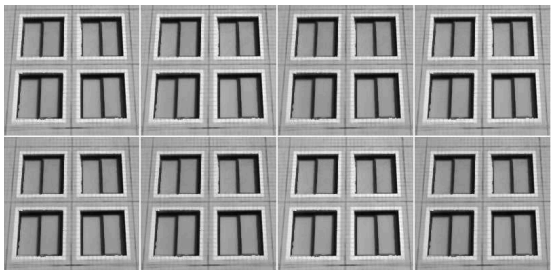


(e) Errors G

Figure 4.5: Robust recovery and alignment by sparse and low-rank decomposition in handwritten digits.

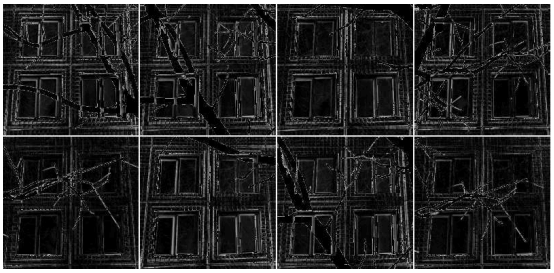
(a) Original homography images A

(b) Aligned images $A \circ \tau$



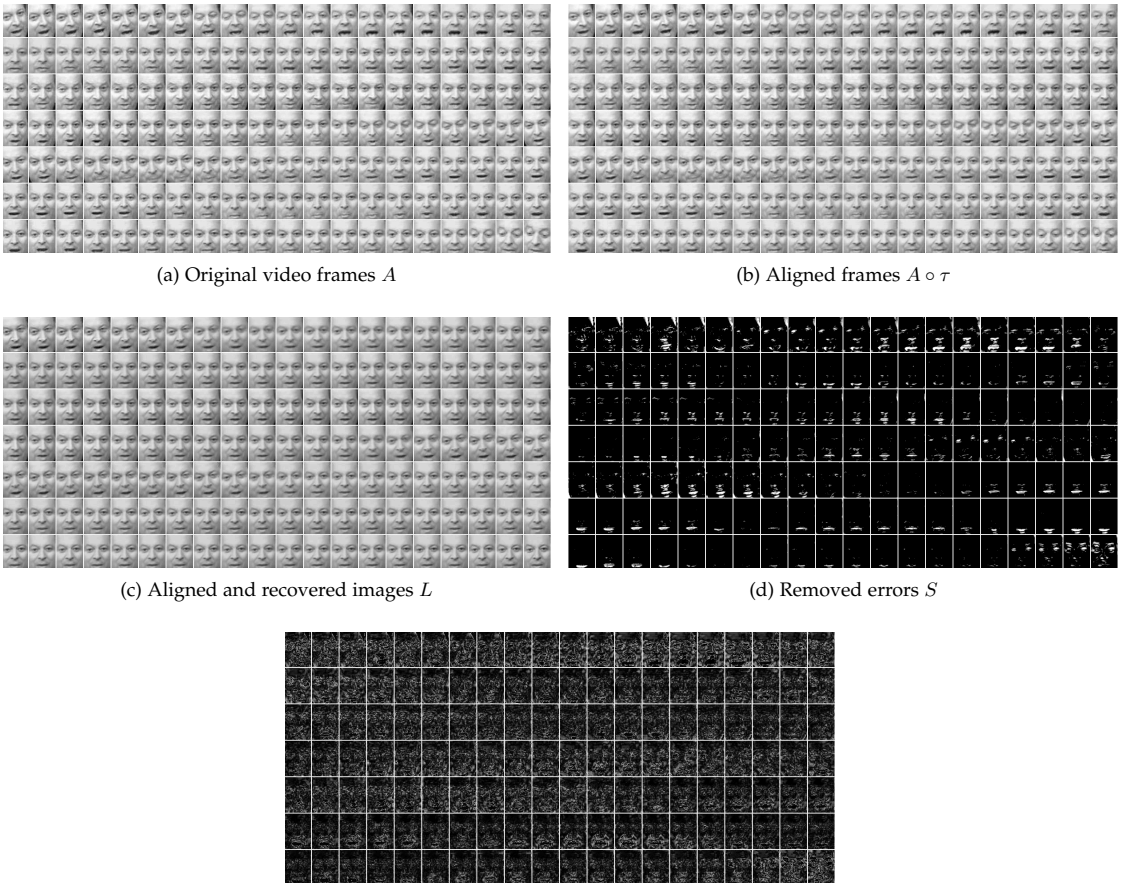
(c) Reconstructed images L

(d) Removed occlusions S



(e) Removed occlusions G

Figure 4.6: Alignment and recovery of planar homographies.



(a) Original video frames A

(b) Aligned frames $A \circ \tau$

(c) Aligned and recovered images L

(d) Removed errors S

(e) Removed errors G

Figure 4.7: Video stabilisation for recovery of object of interest.

Chapter 5

Background Modelling and Foreground Segmentation

In video analysis background subtraction consists of creation of a background model that allows distinguishing foreground pixels. We present a new method in which we regard the image sequence to be made up of the sum of a low-rank background matrix and a dynamic tree-structured sparse matrix. We solve the decomposition task using our approximated Robust Principal Component Analysis method which is extended to handle camera motion and noise. Our contribution lies in dynamically estimating the support of the foreground regions via a superpixel generation step, so as to impose spatial coherence on these regions. Unlike smoothness constraint such as MRF, our method is able to obtain crisp and meaningful foreground regions, and in general, handles large dynamic background motion better. To reduce the dimensionality and curse of scale, we present a variant of our method where we model the background via a Column Subset Selection algorithm, that reduces the order of complexity and hence decreases computation time. Comprehensive evaluation on four benchmark datasets demonstrate the effectiveness of our method in outperforming state-of-the-art alternatives. The findings of this chapter are published in [40], [42], [45], [41], and [47].

5.1 Introduction

Background subtraction can be defined as segmentation of a video sequence into the foreground and the background. It is typically used as a pre-processing step for higher level problems, such as automated surveillance, action recognition, intelligent environments, etc. In the original approach for *background subtraction*, a single static image of the background is subtracted from the current frame, to generate a difference image. If the absolute difference is higher than a threshold, the pixel in question is declared to belong to the foreground; otherwise, it is cast away. This approach performs poorly, as assuming a static never-changing background, in practice is almost never the case. The destiny of the ambiguous pixels that can neither be categorised as foreground nor be let to reabsorb into the background model, is also unknown. Other than a simple thresholding approach, there seems to be no proper mechanism in the current literature by which these pixels are correctly distinguished from others. Background subtraction poses a number of challenges in realistic environments, such as:

- *Presence of noise*, due to low picture acquisition quality, photon noise and varying brightness, thermal images, or low-light sensor automatic adjustments resulting in high noise levels.

- *Illumination changes*, either due to natural gradual daylight or weather changes, or sudden indoor light switch toggles.

- *Background motions or dynamicity*, (trees, water rippling, etc.) whose magnitude can be greater than those of the foreground, but follow no rigid object behavior, shape, or texture.

- *Beginning moving object*, when an object initially in the background moves, both itself and the newly revealed parts of the background called "ghost" are detected.

- *Camouflage*, where a foreground object cannot be discerned from background due to texture and colour similarity.

- *Moved, inserted, or sleeping object*, the transitioning of a background object to the foreground class or vice versa. These objects should not be considered part of their former class forever after the class transition; e.g., a car being parked in the scene, and after a duration of time be considered background, only to later become foreground again when driven off.

- *Camera motion*, which often exhibits itself in Pan-Tilt-Zoom cameras, or mounted CCTV cameras that are subject to wind or vibrations referred to as camera jitter. This movement causes global background motion, which can be considered to be highly correlated global noise. Without a mechanism to handle camera motion the foreground mask would show false detections.

- *Camera automatic adjustments*, modern cameras have autofocus, automatic gain control, automatic white balance, and auto brightness control. These adjustments modify the dynamic in the colour levels between different frames in the sequence.

- *Bootstrapping*, where a training period with no foreground objects is not available. The algorithm must be able to learn the correct background model over time.

- *Shadows*, can be detected as foreground and can come from background objects or moving objects.

- *Foreground aperture*, where slow-moving foreground pixels are absorbed into the background model, when the model adapts too quickly, resulting in a high false negative rate. Also caused by a homogeneously-coloured object moving across the scene, and the change in the interior pixels cannot be detected, thus the entire object may not appear as foreground. An example of this problem can be seen in Figure 5.1.

For a full list of challenges of background subtraction refer to [8].

Figure 5.1: Foreground aperture problem. left and middle: two frames that are 5 frames apart. Right: when a homogeneously-coloured object moves very slowly, the only visible change for the model is the green and magenta regions, therefore the model is blind to the white region.

5.2 Background Modelling and Foreground Segmentation Framework

Addressing the challenges introduced in the previous section, leads to a number of considerations in designing a background model, as well as expected behavior from foreground objects, which in complex applications remains an open problem. *Noise* is generated by image capturing process and small variations in the pixel colours. Here we model *noise* by the residual error of the approximation of background plus foreground. *Illumination changes* are handled to some extent via a robust background model that is capable of adapting itself to global variations of luminance. On account of *dynamic* nature of the background, both the background model and the foreground classification mechanism must be able to correctly classify a range of pixels. These pixels must not reabsorb into the background, since they can violate the model or contaminate regions of the background. Also, they must not be classified as foreground; thus the foreground classification constraints must discern between these pixels and genuine foreground pixels, and therefore, discard them as noise. *Camouflage* on the other hand, is when a foreground object due to its similarity to the background persists absorbing into the background. This effect is interleaved with challenges of noise. Noise can increase the range of values considered to belong to the background, allowing camouflaged objects to remain unde-

tected. Consequently these models suffer a trade-off between a slow-adapting background where noise triggers detection, and a quick-adapting one where camouflaged objects are missed. Noting this challenge, there is a need for two semantic foreground layers, one containing genuine foreground regions, and the other ambiguous and noise-like pixels. Then, the amount to incline toward which layer for detecting foreground objects must be adaptively controlled by a robust mechanism in the model. On the other hand, a desirable background model must be able to learn a variety of modes from the video feed, such that it handles variations in the background, *moved objects*, and noise without compromising its ability to detect camouflaged regions.

To this end we propose to use an approximated form of the *Robust Principal Component Analysis* (RPCA) method for background modelling and foreground segmentation. Given a data matrix containing the frames of a video sequence stacked as its columns, $A \in \mathbb{R}^{m \times n}$, RPCA [17] solves the matrix decomposition problem

$$\min_{L,S} \|L\|_* + \lambda \|S\|_1 \quad s.t. \quad A = L + S, \tag{5.1}$$

as a surrogate for the actual problem

$$\min_{L,S} rank(L) + \lambda \|S\|_0 \quad s.t. \quad A = L + S, \tag{5.2}$$

where in this decomposition, L is the low-rank component corresponding to the background and S is the sparse component containing the foreground part. This formulation promises exact recovery of two matrices L and S with high guarantees, and is generalisable to many problems where data is high-dimensional and a low-dimensional representation is sought. However, in some applications such as background subtraction it performs poorly for two main reasons. Firstly, that the low-rank part will usually yield a matrix with unnecessarily high rank, mainly due to noise and sensor induced corruptions, which is not favourable for a background subtraction application. Secondly, that the sparse part will contain many false alarms (Gaussian noise, ambiguous pixels, and

corruptions) that render the result as obtained, unusable for practical object detection and recognition applications. For these reasons, many RPCA-based methods resort to a final thresholding on S, that would result in less than ideal revealing of foreground support. It is needless to mention that it would be impossible to handle large camera motion, with the formulation above. To overcome these inherent limitations of RPCA for background subtraction and foreground detection, we are interested in the case where we can decompose the matrix A into three components, namely a low-rank part L, a sparse component S, and a residual part E. That is, the target is to decompose A as

$$A = L + S + E, \tag{5.3}$$

where L describes the background of the underlying video sequence, the sparse matrix S mostly contains foreground regions, and the residual error E attempts to capture noise and ambiguous pixels. This model does not assume an exact scene decomposition into background and foreground; it rather addresses an approximate decomposition $A \approx L + S$ [40], [45], [42], [176], with E encapsulating the residual error of the approximation of A by $L + S$. Observe that we expect L to be a genuine low-rank matrix, thus $rank(L) \ll rank(A)$. Moreover, by decomposing all the extra noise that contaminates the background, and storing it into E we are able to reduce the rank of the matrix L beyond what (5.1) is capable of. The L in (5.3) is much more well-suited for background subtraction applications, or in general where lower dimensional models are more desirable.

Background modelling by low-rank approximation has a number of benefits: firstly, a robust estimation of the mostly static regions of the image is guaranteed; secondly, this approximation can partially handle the illumination variations in the background, such as a tree swaying backwards and forward, or water rippling in a lake, traffic light changes that can be modelled by a few modes, or billboards in a street displaying a few images on repeat. Thirdly, low-rank approximation of the background can help distinguish between general motion in the scene, which can be due to camera movement,

and local varying motions caused by moving objects; since background regions obey a single highly correlated motion pattern.

Despite the promising effects of using a low-rank approximation for obtaining the background model, a sparse constraint for foreground objects, is far too limiting. In addition, processing per-pixel basis from the foreground, is not only time-consuming, but also can dramatically affect foreground region detection, if region cohesion and contiguity is not considered in the model. The foreground regions are spatially coherent clusters. Thus, we prefer to detect contiguous regions of various sizes, and then lots of zero entries (regions) in the sparse matrix. With this objective in mind, we propose structured-sparsity inducing norms that are effective in the context of a novel dynamic group structure, by which the natural structure of foreground objects in the sparse matrix is preserved. The dynamicity of group structures is either controlled via a patch-based group selection algorithm, or derived from the natural shape of objects in the scene – by selecting clusters of pixels via the SLIC superpixels [1], and dynamically refining the size of these clusters in an iterative process. This is effective in reducing the *foreground aperture* problem in our experiments.

Because we solve an approximated RPCA problem, it is important to drive the algorithm by means of knowledge of salient regions and the distribution of outliers, so that the algorithm finds the correct solution for the problem at hand. However, a knowledge of the object of interest before even segmenting it makes the problem as one of the many chicken-egg problems in computer vision, as we usually need to segment the scene to recognise the objects in it. So, to identify the foreground objects and their probable size and location, we use an intuitive initialisation step by which the background is encouraged to lean towards the best low-rank approximation of the static parts in the scene, and the sparse part is initialised to take on high probability values for regions of the scene with highest statistical *leverage* scores, similar to a motion-saliency map.

The matrix A can become humongous when processing large or long videos. To alleviate the dimensionality and the curse of scale with an RPCA-based problem, we must

leverage on the fact that such data have in fact low intrinsic dimensionality, e.g., that they lie on some low-dimensional subspace, are sparse in some basis, or lie on some low-dimensional manifold. The simplest and most useful assumption is that the data all lie near some low-dimensional subspace. This is the basis for low-rank approximation, but still does not help with huge matrices. There exists an algorithm named *Column Subset Selection Problem* (CSSP) [14], [123] that selects a handful of the most representative and important columns of a matrix. Assuming that we have a long video of a scene at our disposal with hundreds or even thousands of frames, only a handful of these frames determine a model of the background; the rest will either contaminate the background or will be redundant to process. To this end, we propose to model the background of the sequence using a low-rank approximation from the output of the CSSP algorithm. Following the theoretical recommendations of this algorithm we shall arrive at a near-optimal rank-κ approximation of the matrix A. Not only does this algorithm reduce the complexity and the computation time, but also alleviates the *bootstrapping* challenge, making it possible to still be able to obtain a robust model of the background without needing to observe a clean, foreground-absent frame; i.e., it is possible to complete a matrix from a fraction set of grossly corrupted observations. In addition, it avoids the need to store and process hundreds of video frames, which consume vast amount of memory [152]. This memory consumption problem is particularly noticeable for approaches based on RPCA [178], [176], [66], [101], [40].

In a nutshell contributions of our proposed method are:

- low-rank approximation of the background to accommodate small scene and illumination changes to some extent;
- inducing structured-sparsity in a novel group structure, namely a dynamic block structure and a dynamic superpixel structure;
- insensitivity to foreground object size, as a result of using within-patch normalisation;

- assumption of a noise part in decomposition for reducing false positive pixels (false alarms);
- a *tandem* algorithm for removal of unwanted ghosting effects that persist in background subtraction process, and targets the unascertained prior knowledge of distribution of outliers;
- a dimensionality reduction for RPCA problem via the *Column Subset Selection Problem* that alleviates *bootstrapping*, and reduces computational complexity and cost, and an analysis of the efficacy of this method.

Finally, an exhaustive evaluation using four datasets [152], [15], [91], [160], demonstrating top performance in comparison with the state-of-the-art alternatives is presented.

5.3 Related Work

The background subtraction and foreground detection field is humongous with many surveys available [15], [24], [83], [67], [11], [126], [8]. One of the most prevalently used methods [142] uses a Gaussian mixture model (GMM) for pixel density estimation, followed by a connected components regularisation. Due to its effectiveness in sustaining background variations, a large amount of further developments [180],[69] have been proposed. Another well-known method [37] proposed a non-parametric kernel density estimate (KDE) method for background modelling.

With the advent of deep learning, and the recent success of [86] in the ImageNet [131] image classification challenge, many algorithms based on convolutional neural networks (CNN) have emerged for image classification and segmentation with gold standard performance. Among the most notable recent ones the work of [103] proposed fully convolutional networks for semantic segmentation. The idea is based on adding a fully convolutional layer on top of the CNN features to generate a mask of segmentation output; this is a pioneering work towards extending the already existing classification networks to seg-

mentation. Other works based around the same idea, have emerged each surpassing the accuracy of their predecessor networks such as [85], [130], [102], [3], [22]. Very recently, the work of [72] named Mask R-CNN surpassed these performances by extending the Faster R-CNN framework [129] – which works extremely well for object detection – for pixel-level segmentation in the image, as well as instance labelling of the segmentations. However, all these methods work for single images; obtaining a temporally-consistent segmentation across adjacent frames in a video sequence still remains an open research problem. Moreover, deep learning-based models, and more specifically the supervised class of these models, are heavily data-driven and can only surpass human accuracy in object segmentation only when they are fed millions of hand-annotated examples, and allowed to train for hundreds of hours on expensive GPUs (Graphics Processing Unit). Although with the availability of data, and strategies for low-cost human annotation mining, as well as diminishing cost of GPUs and TPUs (Tensor Processing Unit) the trend of data-driven approaches to solutions of computer vision problems is expected to continue in foreseeable future.

In the recent years, global models such as principal component analysis (PCA) [120] have gained some popularity due to their computational simplicity and effectiveness in camera shake. They attempt to model the background as a low dimensional subspace of the vectorised input, with the foreground identified as outliers. In practice such approaches have struggled, due to high computational requirements and limited capability to deal with many common problems, e.g., camouflage. Recent variants have resolved part of these issues, notably [71] proposed a non-SVD based fast solution. However, still no considerations of the spatial distribution of outliers was considered. In an effort to incorporate such prior an MRF-based solution [178] has been proposed. But the result of imposing such smoothness constraint (even with the discontinuity preserving prior such as those based on Potts model) is that the foreground regions tend to be over-smoothed; as an example, the details in the silhouette of hands and legs of a moving person is sacrificed in favour of a more compact blob.

Our idea is established in the so-called structured-sparsity or group-sparsity measures to incorporate the spatial prior. Structural information about non-zero patterns of variables have been developed and used in sparse signal recovery, and many approaches have been applied to these problems successfully, such as Lattice Matching Pursuit (LaMP) [18], Dynamic Group Sparsity (DGS) recovery [77], Bayesian Robust Matrix Factorisation (BRMF) [159],and the Proximal Operator using Network Flow (ProxFlow) [112]. However, the majority of related methods [36], [143], [58] typically assume that the block structure and its location is known or will suffer in *regularisation* or *bootstrapping*. To lift up some difficulties, Rosenblum et al.'s [172] method instead detects the block size and location by iteratively alternating between updating the block structure of the dictionary and updating the dictionary atoms to better fit the data. Nevertheless, both the number of blocks and the maximal block size are assumed to be known. In [94], [125] the sparsity structure is estimated automatically, however parameter tuning is required in [77] to control the balance between the sparsity prior and the group clustering prior for different cases, and both methods need a clean background to train backgrounds for sequences. The authors of [58] used a two-pass RPCA framework, in which the first pass determines a saliency map is generated that corresponds to locations of the outliers, and then the second pass uses the 4×4 salient blocks in the image, to favour spatially contiguous outliers. In another effort [101] used a group sparse structure, in which overlapping groups of 3×3 pixels in an 8×8 region of an image are used in conjunction with a maximum norm regularisation to take into account the spatial connection of foreground regions. In a recent work [79] a superpixel-based max-norm matrix decomposition approach has been proposed, in which homogeneous static or dynamic regions of image are classified as a graph partitioning problem, via Generalised Fused Lasso (GFL). In contrast, our method does not assume a prior size or location or structure for sparsity, and dynamically updates these to best fit the natural object shape in the scene, without a separate training phase.

The vast majority of approaches, including ours, use the colour information of pixels

directly. We have noticed the most reliable information source at least for our approach is still the pixel colour information. Additionally, only foreground/background classification is provided by the presented approach – other approaches may mark regions as being in shadow or use other labelling strategies. Moreover, post-processing techniques such as eroding then dilating are common in background subtraction. However, to provide a meaningful evaluation of our algorithm we refrain from performing any post-processing on our results.

5.4 Approximated RPCA for Background Modelling and Foreground Segmentation

As discussed in the previous section, our proposed approach is based on an approximated RPCA process, that takes advantage of natural structure of objects in the scene. In our model a series of structured-sparsity inducing norms are defined which act in a tree structure that is a representation of the scene components. Similarly to (2.6) the approximated decomposition problem stated in (5.3) can be solved by minimising the decomposition error

$$\min_{L,S} \|A - L - S\|_F^2 + \lambda\|S\|_1 \quad s.t. \quad rank(L) \leq r \ll rank(A), \tag{5.4}$$

where $\|\cdot\|_F$ is the Frobenius norm. In the Frobenius norm, the set of feasible solutions is restricted to matrices L that have a rank smaller than or equal to r. It means that if r is much smaller than $min(m, n)$, the solution for L is necessarily a low-rank matrix. λ is a tuning parameter set at a value that helps recovering all genuine foreground regions. We find that using $\lambda = 3/\sqrt{max(m, n)}$ (where $m \times n$ is the dimensions of A) is adequate to identify all foreground regions in our test data. The choice of λ is justified by observations in our experiments, where λ controls a good trade-off between the sparsity of $S + E$ and structured-sparsity of S. The matrix E contains the residual error of the

approximation of A by $L+S$. The entries of this matrix can be very large in magnitude, but random and scattered, exhibiting noise, and showing no structured shape in the sparsity domain. Therefore, they should neither remain in the foreground as they will trigger many false positives and pollute the foreground model, nor be able to get absorbed into the background model and increase its rank. Our tree structured-sparsity inducing norms ensure the former case, and the robust low-rank approximation will ensure the latter. Most background subtraction methods suffer from this kind of contamination polluting their foreground model, and consequently resort to a final thresholding step or post-processing once the foreground support is calculated. Not only is this redundant but also counter-intuitive, as the source of the pollution can be eliminated to a practical degree during the iterative minimisation; therefore, decomposing the original matrix into an additional noise term E proves to be effective.

It seems as if the variability of $rank(L)$ along with controllability of S with λ would already provide solutions to three of the challenges in background modelling, namely *noise*, *illumination changes*, and *dynamic backgrounds*. But that is not enough to guarantee crisp and meaningful foreground segmentation. To address camera-induced background motion which is present in most scenes, we incorporate a step into the minimisation process, in which we estimate some transformation that can describe the general motion in the scene. The robustness and efficacy of this step has been proven in [124], [178] where the estimated background model is assumed to be under some transformation described by motion parameter vector τ. Therefore, no local motion in the scene will affect the estimation of these motion parameters.

Next, to address *foreground aperture* problem, we employ structured-sparsity inducing norms in the context of tree-structured groups. We exploit the natural shape of objects in the scene which best describe the location and distribution of outliers. Validated by our experiments, this proposition significantly reduces the *foreground aperture* problem, and produces object segmentation in coherent clusters. Another merit of the proposed structured sparsity framework is its ability for generalisation to many different

scenes, as shown in our experiments. Since this framework is carried out coarsely over regions with no foreground objects and very vigorously on regions hinting presence of foreground, the amount of computation is significantly reduced.

Thus far, while the low-rank formulation of the background matrix is generally effective in absorbing many natural variations in the background, the full power of the approximated RPCA framework to achieve accurate decomposition can only be achieved if somehow a subtle mechanism can handle the scale issue and conceive the expected position and extent of foreground motions. It is evident that there is no single λ than can achieve the cleanest separation of foreground and background regions, if no prior knowledge of foreground outliers is provided to the approximated RPCA algorithm.

We will discuss in the upcoming sections a *tandem* algorithm acting as an initialisation that supports the main drive in the decomposition. Despite its striking simplicity it gives surprisingly good effects in removing significant number of non-stationary background points that are deposited in the outlier matrix, as well as eliminating a faint trace of the foreground motion which leaves a *ghost*-like presence in the background matrix.

5.5 Modelling with Structured-Sparsity Inducing Norms

We employ structured-sparsity inducing norms in the context of tree-structured groups, where the natural shapes of objects in the scene are exploited to best describe the location and distribution of foreground regions.

We propose sparsity-inducing norms that can incorporate prior structures on the support of the errors such as spatial continuity. We essentially consider a special case to the following problem

$$\min_{rank(L)\leq r,S} \|A - L - S\|_F^2 + \lambda\psi(S), \tag{5.5}$$

with the regulariser $\psi(\cdot)$ on S chosen to be $\|\cdot\|_{2,1}$. $\ell_{2,1}$-norm is a group sparsity inducing norm. Clearly, the ℓ_1-norm regularisation treats each entry (pixel) in S independently.

It does not take into account any specific structures or possible relations among subsets of the entries. While in background subtraction scenarios, outliers (objects in the scene) normally have the structural properties of spatial contiguity and locality. Indeed, as reported in [81], ℓ_1-norm performs better in case of random pixel corruption than contiguous occlusion. Unfortunately the latter case is actually closer to practical situations in background subtraction. Hence, our choice of $\ell_{2,1}$-norm assures selecting the discriminative input features shared across multiple binary predictors.

Motivated by recent advances in structured sparsity [80], [81], to induce more diverse and sophisticated sparse error patterns, we consider structured sparsity-inducing norms that involve overlapping groups of variables. Although it still assumes pre-defined group structures, the overlapping patterns of groups and norms associated with the groups of variables allow to encode much richer classes of structured sparsity. In this work, we consider a tree-structured sparsity-inducing norm. It involves a hierarchical partition of the m variables in S into groups, as shown in Figure 5.2. The tree is defined in a way that leaf nodes are singleton groups corresponding to individual pixels, and internal nodes/groups correspond to local patches of varying size. Thus each parent node contains a hierarchy of child nodes that are spatially adjacent to each other and constitute a local part in the sparse image S. As illustrated in Figure 5.2, when a parent node goes to zero all its descendants in the tree must go to zero. Consequently, the non-zero or support patterns are formed by removing those nodes forced to zero. This is exactly the desired effect of structured sparse patterns.

We can represent a scene using a tree structure by subdivision. In such a tree structure each child node is a subset of its parent node and the nodes of the same depth level do not overlap. Denote $\mathcal{G}$ as a set of groups from the power set of the index set $\{1, \ldots, m\}$, with each group $G \in \mathcal{G}$ containing a subset of these indices. The aforementioned tree-structured groups used in this chapter are formally defined as follows: A set of groups $\mathcal{G}$ is said to be *tree-structured* in $\{1, \ldots, m\}$ if $\mathcal{G} = \{\ldots, G_1^i, G_2^i, \ldots, G_{b_i}^i, \ldots\}$ where $i = 0, 1, 2, \ldots, d$, d is the depth of the tree, $b_0 = 1$ and $G_1^0 = \{1, 2, \ldots, m\}$, $b_d = m$ and

correspondingly $\{G_j^d\}_{j=1}^m$ are singleton groups. Let G_j^i be the parent node of a node $G_{j'}^{i+1}$ in the tree, we have $G_{j'}^{i+1} \subseteq G_j^i$. We also have $G_j^i \cap G_k^i = \emptyset, \forall i = 1, \ldots, d, j \neq k, 1 \leq j, k \leq b_i$. Similar group structures are also considered in [81], [99]. With the above notation, a general tree-structured sparsity-inducing norm can be written as

$$\psi(S) = \sum_{i=0}^{d} \sum_{j=1}^{b_i} w_j^i \| S_{G_j^i} \|_{2,1}, \tag{5.6}$$

where $S_{G_j^i}$ is a vector with entries equal to those of S for the indices in G_j^i and 0 otherwise. w_j^i are positive weights for groups G_j^i. It is chosen as $w_j^i = 1/max(A_{G_j^i})$ to overcome sensitivity of the regularisation scheme to illumination variance across patches. This is crucial as using the same λ for all the patches in the scene will usually favour the most prominent features (in this case the illumination variations with largest magnitude). By normalising each patch with a weight associated with the highest colour variation in that patch, this issue is largely subsided; and as such the *camouflaged* objects will have a higher chance of being detected.

For the $\ell_{2,1}$-norm – which is the ℓ_1-norm of the vector formed by taking the ℓ_2-norm of a matrix – it is the maximum value of pixels in a group that decides if the group is set to non-zero or not, and it does encourage the rest of the pixels to take arbitrary (hence close to maximum) values. The effectiveness of this choice is corroborated with empirical evidences in [65], [118], [98], [117], [170]. This norm definition promotes sparse error patterns more consistent to practical object detection than standard ℓ_1-norm used widely in the literature for this kind of problem.

5.5.1 Robust foreground detection via structured sparsity

In this section, we use the defined structured sparsity-inducing norms of the last section to replace the traditional ℓ_1-norm for modelling the foreground regions in robust background subtraction. Thus, the objective function in the optimisation program is modified to the

following

$$\min_{rank(L)\leq r,S} \|A - L - S\|_F^2 + \lambda \sum_{i=0}^{d} \sum_{j=1}^{b_i} w_j^i \|S_{G_j^i}\|_{2,1} \tag{5.7}$$

where λ is a parameter controlling the trade-off between sparsity of $S + E$ and structured sparsity of S. To solve (5.7) we use an alternating minimisation procedure. This kind of iterative linearisation has a long history in gradient algorithms. We proceed by minimising the function for two parameters L and S one at a time until the solution reaches convergence; that means solving two reduced problems, each being minimised independently from one another

$$L^t = \underset{rank(L)\leq r}{\arg\min} \|A - L - S^{t-1}\|_F^2 \tag{5.8}$$

$$S^t = \arg\min_S \|A - L^t - S\|_F^2 + \lambda \sum_{i=0}^{d} \sum_{j=1}^{b_i} w_j^i \|S_{G_j^i}\|_{2,1} \tag{5.9}$$

Both these subproblems have non-convex constraints. Their global solutions L^t and S^t exist. In particular, the two subproblems can be solved by updating L^t via singular value hard thresholding of $A - S^{t-1}$ [176], and updating S^t via our structured-sparsity inducing norms with a soft-thresholding with λ. The penalty term in (5.9) assures the structured-sparsity of S w.r.t. the defined tree-structured groups. The most notable difference in update of S^t with [176] is the introduction of the novel structured-sparsity inducing norms, with a penalty term similar to [81], that assures recovering correct foreground regions based on a sparsity assumption. In [176] this update is controlled with a non-convex cardinality constraint and then entry-wise hard-thresholding that is computationally expensive.

5.5.2 Defining tree-structured groups in meaningful regions

A meaningful structured-sparse solution, is the one that is best able to take into account the natural shape and structure of objects in the scene. There is a need for some mechanism that describes each tree-structured group $\psi(\cdot)$. Each group must take into account connected components belonging to a semantically or texturally connected region. For example, a region of pixels with the same colour and texture belonging to part of an object (a wheel of a car) must be assigned to a single group. The structured sparse inducing framework defined in the previous section can then be used within the group class to decide whether it belongs to foreground or must be classified as background.

A trend in recent literature has been shifting towards a very common approach in video coding technology, where the test image is divided into square-shaped regions of pixels called blocks, with pre-determined sizes. To get even more elaborate with this sectioning, each block can be further divided into smaller blocks each time halving the size of the block. This can be done until a block of size 1 (a single pixel) is reached; this is called the quad-tree decomposition. This approach is not very complex and can be implemented with low order of computation in the framework we described in the previous sections. Despite its simplicity, it has an inherent limitation that the location of the blocks is always fixed in the image, and therefore it would not be a very flexible solution to all scenarios. To compensate for this limitation, the blocks can be defined to be overlapping. This is achieved by the overlapping tree-structure $\psi(\cdot)$ in (5.6). Figure 5.2 shows an example structure of such blocks. In this example a region of 8×8 pixels is chosen as a group. If there are no elements with large magnitude in this region, the sparsity-inducing norms will classify the whole region to background; otherwise it is divided into 4 smaller regions of 4×4 pixels. Similarly, each of the smaller regions are put to the test of sparsity-inducing norms, and the regions belonging to background are left-off, while the regions hinting foreground elements are divided into 4 smaller regions once again. This is done until a singleton group (a single pixel) is reached. We call this procedure *induction*, *division*, and *discarding*. There are two immediate

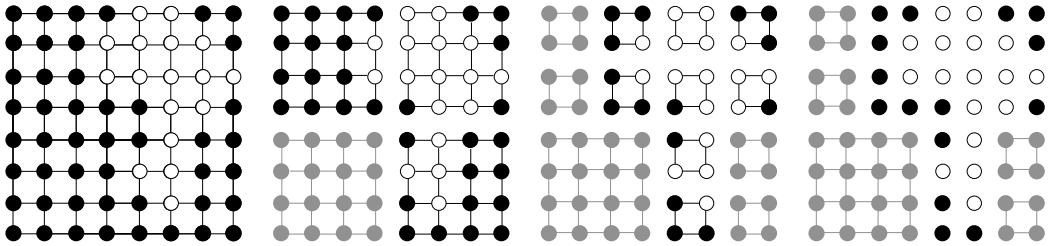


Figure 5.2: Dynamic group sparsity induction, division, and discarding procedure of DBSS. A region is divided into smaller regions, the ones indicating foreground presence are kept and divided for further induction, whilst grayed-out regions are immediately discarded as they contain no foreground.

benefits from defining such a block structure: firstly, the amount of computation needed for classification is lowered, as classifying larger regions to background is much faster compared to single pixel assignment, while for blocks containing foreground objects the subdivisions will allow more meticulous investigation in these regions. Large region classification can be safely done in our model; this is the direct impact of our sparsity-inducing norms definition, since despite other RPCA-based methods our algorithm is not sensitive to the size of the region in question. Secondly, the recursive division of regions down to one pixel will result in a very crisp and well-defined foreground segmentation as shown in Figure 5.3. We refer to this approach in this chapter as *DBSS* model, short-hand for *Dynamic Block Structured Sparse*.

Depth of each tree in this model is set to $d = 3$ and $m = 64$, therefore $\mathcal{G} = \{\dots, G_1^i, G_2^i, \dots, G_{b_i}^i, \dots\}$ where $i = \{0, 1, 2, 3\}$, $b_0 = 1$ and $G_1^0 = \{1, 2, \dots, 64\}$, $b_d = 64$ and correspondingly $\{G_j^d\}_{j=1}^{64}$ are singleton groups. The general tree-structured sparsity-inducing norm becomes

$$\psi(S) = \sum_{i=0}^{3} \sum_{j=1}^{b_i} w_j^i \|S_{G_j^i}\|_{2,1} \tag{5.10}$$

As mentioned before, DBSS bears two limitations that the size and location of the blocks need to be set in advance, and it is hard to see how each block is adapting its shape to the natural structure of objects in the scene. Motivated by these limitations, we propose a new group structure, in which the sparse part derives its structure from

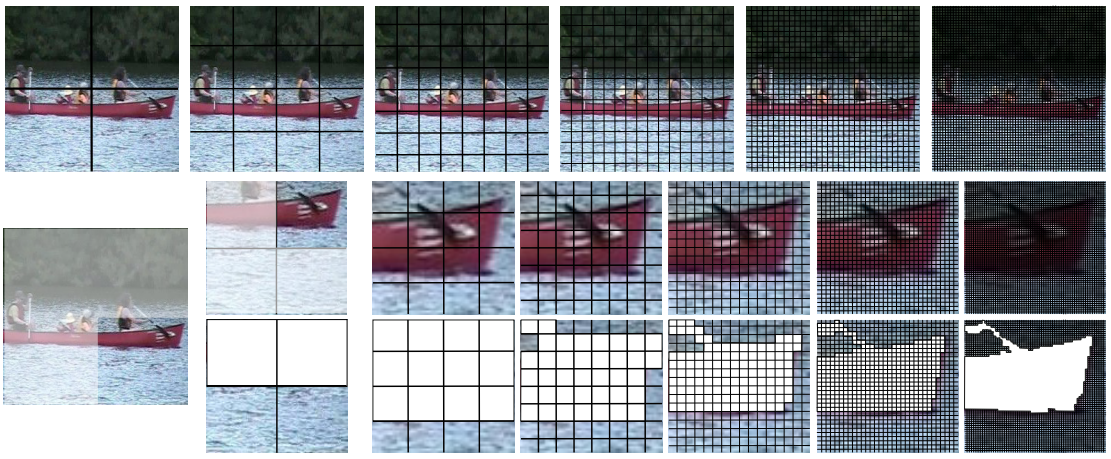


Figure 5.3: The tree-structured sparsity constraints yield accurate and crisp foreground segmentation in DBSS.

the natural object structure in the scene. In a test image, the scene can be classified into multiple *superpixels*. Recent advances in image segmentation, have made many superpixel algorithms available, that promise state-of-the-art ability with respect to adherence to image boundaries, speed, memory efficiency, and segmentation performance. A good superpixel must obtain perceptually meaningful atomic regions, which can be used to replace the rigid structure of the pixel grid. Moreover, as these results will be used as a pre-processing step in our foreground detection framework, they should be fast to compute, memory efficient, and simple to use. Also, in our segmentation scenario, superpixels should both increase the speed and improve the quality of the results.

We therefore, adopt the *simple linear iterative clustering* (SLIC) algorithm based on the empirical comparison of six state-of-the-art superpixel methods [1]. SLIC adapts k-means clustering to generate superpixels, and is freely available[1]. By default, the only parameters of the algorithm are the desired number of approximately equally-sized superpixels, and a compactness factor controlling adherence of each superpixel region to object boundaries. Figure 5.4 shows a few examples of superpixels in our test data. The number of superpixels in the upper left of each image is 100 superpixels, 500 in the middle, and 2000 in the lower right. It seems that for our test images, 800 superpixels

[1] `http://ivrl.epfl.ch/research/superpixels`

are sufficient to adhere well to all object boundaries.

Once the superpixels are obtained in the pre-processing step, the same procedure for structured sparsity inducing norms is applied to groups, that are this time each superpixel region in the test image. For recursive division however we cannot follow the naïve recursive block division of DBSS. We have adapted SLIC to be able to dynamically divide each superpixel region into approximately equal-sized smaller superpixels.

SLIC superpixels might actually give fewer number of equally-sized superpixels per region than specified, but not more. We therefore, feed each rectangular region of the image encompassing each initial superpixel into SLIC, taking note of the surrounding coordinates in the rectangle that do not belong to the superpixel. These coordinates are discarded at the end of this process. In this step each rectangular region is divided into 4 equally-sized smaller superpixels. The surrounding coordinates that do not belong to the superpixel can be large, and be labelled as additional superpixels in this division. Therefore, we calculate how many of our yielded superpixels now lie on those regions. If the number of yielded superpixels that lie on those regions is less than or equal to 2, that means the initial superpixel has been successfully divided. Otherwise, if this number is greater or equal to 3, that means that no useful superpixel division was acquired. In this case we simply divide the region into 4 smaller equally-sized rectangular regions, and then feed those smaller regions into SLIC for superpixel division. This process can be performed in parallel for all the rectangular regions of the image, and therefore is efficient. The same procedure is performed again for the obtained smaller superpixels. Our experiments have shown that at this depth (after 3 divisions) the classification can be performed without having to perform any further divisions, as the regions are both small enough to safely discard non-foreground regions, and large enough to crisply classify all foreground objects in the scene with fine details correctly. We denote this model as *DSPSS* short for *Dynamic SuperPixel Structured Sparse*.

Figure 5.5 shows a simplified example of the above process, where each initial superpixel region is divided into 4 smaller superpixels that best adhere to object boundaries.

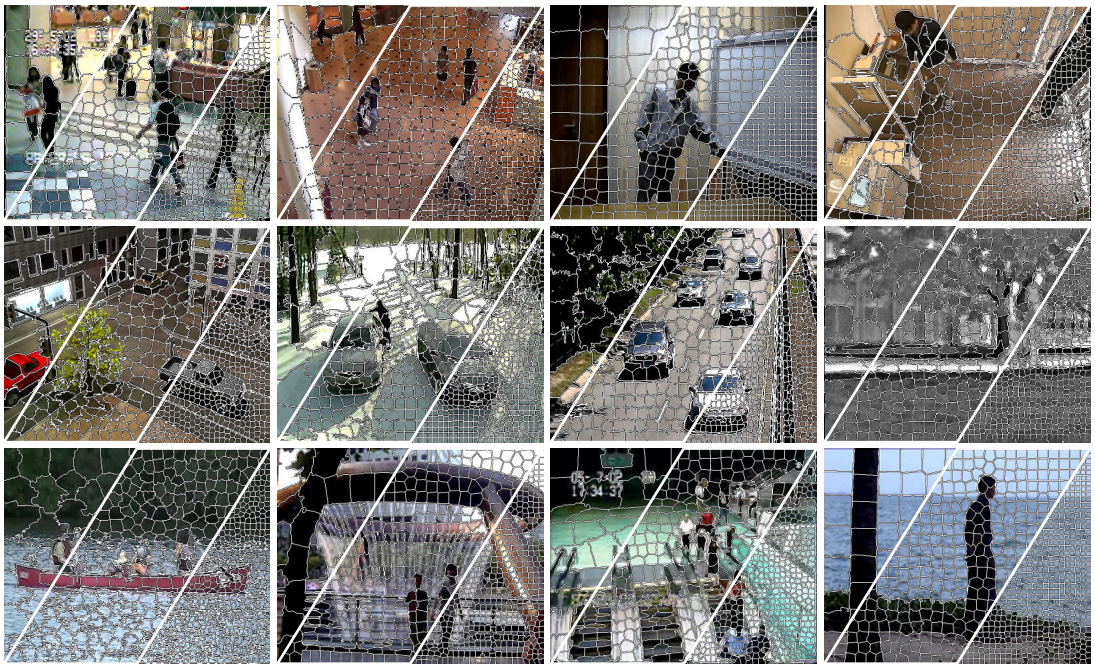


Figure 5.4: Superpixel division in sample data. The number of superpixels in the upper left of each image is 100 superpixels, 500 in the middle, and 2000 in the lower right. It seems that for our test images, 800 superpixels are sufficient to adhere well to all object boundaries.

These smaller superpixels are further divided into 4 regions, again and again. Similarly the parameters for the tree-structured sparsity-inducing norm $\psi(S)$ are defined as follow. Depth of each tree in this model is $d = 3$ and $m = \mathcal{M}$ is dynamically decided by SLIC, since it depends on image size, and the natural shape of the objects in the scene. Therefore $\mathcal{G} = \{\dots, G_1^i, G_2^i, \dots, G_{b_i}^i, \dots\}$ where $i = \{0, 1, 2, 3\}$, $b_0 = 1$ and $G_1^0 = \{1, 2, \dots, \mathcal{M}\}$, $b_d = \mathcal{M}$ and correspondingly $\{G_j^d\}_{j=1}^{\mathcal{M}}$ are the smallest superpixel groups.

5.6 Robust Image Alignment

So far we have assumed that the images in matrix A are well aligned. Precise alignment is crucial for success of sparse representation based background subtraction methods – in fact, good alignment is important for any recognition task. However in practical cases, most video sequences exhibit large amount of camera-induced motion in the background,

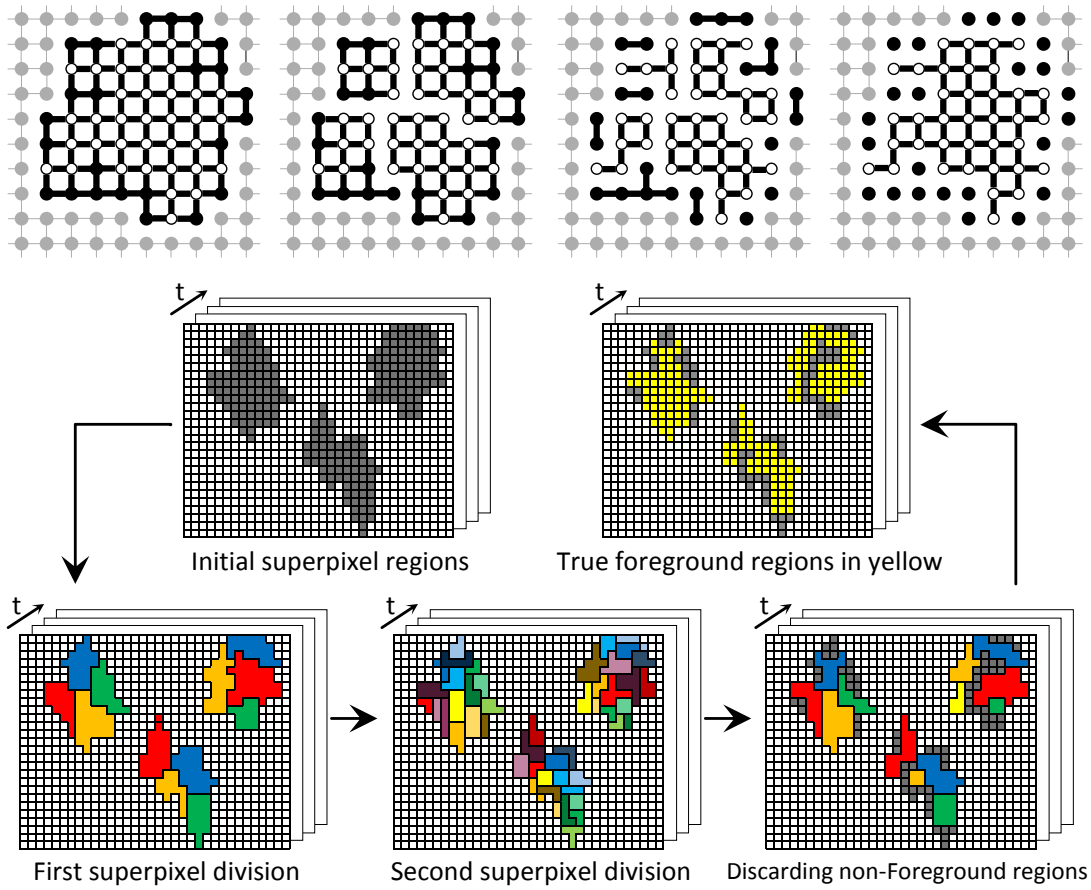


Figure 5.5: Tree-structured groups in sparsity induction, division, and discarding procedure in superpixel regions for DSPSS. This is the same procedure as the DBSS with the exception that the size and location of groups are not known and change from one frame to next.

so that the above assumed linear model no longer holds. In the context of practical background subtraction, A' can be related to A by $A' = A \circ \tau$, where τ stands for some transformation in the image domain (e.g., 2D affine transformation for correcting misalignment, or 2D projective transformation for handling some perspective change). The objective thus becomes to find the correct τ so that after transformation the obtained A from A' can be represented linearly by the training images. The assumption of sparsity itself provides a strong cue for finding the deformation τ.

Suppose A is an observed matrix that is not in register with the training images $\{\mathbf{I}_k\}_{k=1}^{n}$. To recover well-aligned images $A' = A \circ \tau$ such that they can be readily

used for robust background subtraction we propose to solve the following optimisation problem to seek the correct transformation τ and sparse errors S

$$\begin{aligned} \min_{rank(L)\leq r,S,\tau} & \|A\circ\tau - L - S\|_F^2 + \lambda\psi(S) \\ s.t. \quad & A\circ\tau = L + S + E, \end{aligned} \tag{5.11}$$

where each frame in A is sequentially aligned to each frame $\mathbf{I}_k$ instead of the whole training set $\mathbf{I}$, mainly due to the difficulty of optimisation associated with the later case, as discussed in [81]. As an extension to the problem (5.7), based on our structured sparsity, we formulate the alignment problem as the following optimisation objective

$$\min_{rank(L)\leq r,S,\tau} \|A\circ\tau - L - S\|_F^2 + \lambda\sum_{i=0}^{d}\sum_{j=1}^{b_i} w_j^i \|S_{G_j^i}\|_{2,1} \tag{5.12}$$

The problem (5.12) is a difficult, non-convex optimisation problem. Fortunately, we can find a good initialisation by pre-aligning all frames in the sequences to the middle frame, before the main loops of minimisation. The pre-alignment is done by the robust multiresolution method proposed in [119]. This practice is successful in most cases given that a drastic scene change does not occur in the sequence. We can then solve (5.12) by repeatedly linearising about the current estimate of τ, and seeking a deformation step $\Delta\tau$ [124]. In other words, at each iteration, we update τ by a small increment $\Delta\tau$ and linearise $A\circ\tau$ as $A\circ\tau + J\Delta\tau$, where J denotes the Jacobian matrix $J = \frac{\partial A}{\partial\tau}$. Thus, τ can be updated via the following minimisation

$$\tau^t \leftarrow \tau + \arg\min_{\Delta\tau} \|A\circ\tau - L^{t-1} - S^{t-1} + J\Delta\tau\|_F^2 \tag{5.13}$$

The minimisation over $\Delta\tau$ in (5.13) is a weighted least-squares problem that has a closed-form solution. In practice, the update of τ for each frame can be done separately since the transformation is applied on each image individually. Thus the update of τ is efficient. We empirically observe that when A' contains large variations such as

Algorithm 1 Pseudo-code for DBSS and DSPSS with background motion parameter estimation

1: **Input:** A, *rank*, λ, ϵ, *maxIter*
2: **Output:** S, L, E, τ
3: *Standard initialisation:* $\tau^0 = 0$, $L^0 = A$, $S^0 = 0$
4: **while** $\|A \circ \tau^t - L^t - S^t\|_F^2 / \|A\|_F^2 > \epsilon$ or $t <$ *maxIter* **do**
   1) Form the matrix $A \circ \tau$ calculating the parameters τ_i^t that infer the mapping that transforms the column vector A_i to the i-th column vector of the matrix $L^{t-1} + S^{t-1}$.
   2) Calculate $L^t = \sum_{i=1}^{rank} \sigma_i U_i V_i^T$ where $\text{svd}(A \circ \tau^t - S^{t-1}) = U \Sigma V^T$.
   3) Calculate $S^t = \mathcal{P}_\lambda(\psi(A \circ \tau^t - L^t))$ where $\mathcal{P}_\lambda(x) = \text{sign}(x) \max(|x| - \lambda, 0)$.
   4) Calculate the residual noise $E = A - L - S$.
5: **end while**

background being occluded or hidden behind foreground objects, our model is much better than that in [40], [39] for background subtraction and foreground detection as reported in our experiments in Section 5.10.4. Similar to before, we then proceed by minimising the function for two parameters L and S one at a time until convergence

$$L^t = \underset{rank(L) \leq r}{\arg\min} \|A \circ \tau^t - L - S^{t-1}\|_F^2 \tag{5.14}$$

$$S^t = \arg\min_S \|A \circ \tau^t - L^t - S\|_F^2 + \lambda \sum_{i=0}^{d} \sum_{j=1}^{b_i} w_j^i \|S_{G_j^i}\|_{2,1} \tag{5.15}$$

A summary of DBSS and DSPSS methods is described in Algorithm 1; the operator ψ determines which algorithm is used.

5.7 Convergence of the Iterative Process

The sequence of values of the objective function $\|A \circ \tau^t - L^t - S^t\|_F^2 + \lambda\psi(S^t)$, $t = 1, 2, \ldots, p$ produced by the iterative process is monotonically decreasing for a fixed λ converging to a local minimum. The proof is similar to the convergence arguments used by theorem 1 in [176]. The main difference is the addition of a third optimisation problem (which involves the parameters of the motion model) that also has a closed-form solution and the values of the sequence are monotonically decreasing in each step.

5.8 Tandem Approximated RPCA for Removing Ghosting Effects

In this section we propose the *tandem* approximated RPCA where just like a tandem bicycle the front drive (main minimisation loop) is supported by the back pedaling power (initialisation). This proposition involves an initialisation step before the actual optimisation takes place. It is different from algorithms that require a two-pass optimisation [58], where the optimisation is twice performed to refine results. This is rather expensive in an RPCA framework; instead, we strategically initialise the variables such that we gain even better results. This modification will introduce a prior knowledge of the spatial distribution of the outliers to the model. The direct impact of this modification to the RPCA algorithm is faster convergence. The indirect impact is how it alleviates a persisting problem in background subtraction algorithms, called *"ghosting"* effect. The *ghosts* are either parts of the foreground object that remain in the background model, or parts of the background that leak into the foreground. The main reasons causing these artifacts are: an object moving slowly, or remaining inactive for some period of time, or when the foreground object obscures part of the background during the training period. With current RPCA-based optimisations the ghosts usually persist during the iterative process; this can be seen in Figure 5.9. The optimisation problems described in Sections 5.5 and 5.6 are solved by iterative procedures that need to be initialised using starting values for the matrices L, S, and τ. Algorithm 1 starts the iterative process with a standard (naïve) initialisation of $L^0 = A$, $S^0 = 0$, and $\tau^0 = 0$. The rank-r matrix that is the nearest to the matrix A is a low-rank matrix that gives a good first approximation for the static part of the sequence but some parts of the moving objects remain in this rank-r matrix. Hence we propose to construct a matrix S^0 whose columns contain only the more salient part of the difference between A and L^0, where L^0 is the rank-r matrix approximation of the matrix A. This difference matrix $\mathcal{S} = A - L^0$ will contain a sketch of the moving objects in the scene, and therefore is a good initial approximation that contributes to the non-uniformity of the structure of the matrix. We adopt the statisti-

cal *leverage* scores to measure the importance of the columns of the difference matrix. These scores can be regarded as a pseudo-motion saliency map. Let the i-th column of the matrix to be a linear combination of the orthonormal basis given by the left singular vectors of the matrix $\mathcal{S}^i = \sum_{r=1}^{\varrho} \sigma_r U_r V_r^i$, $i = 1, \ldots, \eta$ where U_r is the r-th left singular vector, V_r^i is the i-th coordinate of the r-th right singular vector, and ϱ is the rank of the matrix $\mathcal{S}$. As the matrices $\mathcal{S}_j$ are approximations of the frames containing the moving objects, they can be considered as approximations to low-rank matrices. It implies that one can assume

$$\mathcal{S}_j^i \approx \sum_{r=1}^{\rho} \sigma_r U_r V_r^i, \quad \rho \ll \varrho \tag{5.16}$$

Note that any two columns i_1 and i_2 differ only by $\sum_{r=1}^{\rho} V_r^{i_1}$ and $\sum_{r=1}^{\rho} V_r^{i_2}$. Then these terms can be used to measure the importance or contribution of each column to the matrix. The normalised statistical leverage scores [111] of the i-th column of matrix $\mathcal{S}_j$ is defined as

$$\ell_i = \frac{1}{\rho} \sum_{r=1}^{\rho} V_r^i, \quad i = 1, \ldots, \eta, \tag{5.17}$$

where η is the number of columns of each frame of the sequence. The sub-index j is removed to help understanding this expression. Leverages have been used historically for outlier detection in statistical regression but recently they have been used to give column (or row) order of the amount of motion saliency in a specific part of the image. The vector ℓ_i is a probability vector, i.e. $\sum_{j=1}^{\eta} \ell_i^j = 1$ and $\forall j \in [1, \eta], \ell_i^j \geq 0$. Therefore, the columns of each matrix $\mathcal{S}_j$ with leverages greater than $\frac{1}{\eta}$ are the more important columns. So the columns of the initial approximation S^0 contain only the more important columns of the matrices $\mathcal{S}_j$, $j = 1, \ldots, n$. Consequently, the less salient parts of the image are not included in the initialisation of the sparse part, making the iterative process faster to converge, yielding more stable results, and increasing the segmentation accuracy.

$$S_j^{0^i} = \begin{cases} \mathcal{S}_j^i, & \ell(\mathcal{S}_j^i) \geq \frac{1}{\eta} \\ 0, & \textit{otherwise} \end{cases} \tag{5.18}$$

Algorithm 2 Pseudo-code for DBSS and DSPSS with background motion parameter estimation and Tandem initialisation

1: **Input:** A, *rank*, λ, ϵ, *maxIter*
2: **Output:** S, L, E, τ
3: *Tandem initialisation:* $\tau^0 = 0$, $L^0 =$ rank-r approximation of A, $S^0 = A - L^0$
4: **while** $\|A \circ \tau^t - L^t - S^t\|_F^2 / \|A\|_F^2 > \epsilon$ or $t <$ *maxIter* **do**
   1) Form the matrix $A \circ \tau$ calculating the parameters τ_i^t that infer the mapping that transforms the column vector A_i to the i-th column vector of the matrix $L^{t-1} + S^{t-1}$.
   2) Calculate $L^t = \sum_{i=1}^{rank} \sigma_i U_i V_i^T$ where $\text{svd}(A \circ \tau^t - S^{t-1}) = U \Sigma V^T$.
   3) Calculate $S^t = \mathcal{P}_\lambda(\psi(A \circ \tau^t - L^t))$ where $\mathcal{P}_\lambda(x) = \text{sign}(x) \max(|x| - \lambda, 0)$.
   4) Calculate the residual noise $E = A - L - S$.
5: **end while**

In Figure 5.9 we have shown the effect of the tandem initialisation in our model, with comparison to other RPCA-based algorithms. The ghost effects are visible in foreground parts in the forth to sixth columns of this figure, which in turn contaminate the background model in the eighth to tenth columns in other RPCA-based methods. A summary of DBSS and DSPSS methods is described in Algorithm 2; similar to before, the operator ψ determines which algorithm is used. To initialise values for the matrices L and S in both DBSS and DSPSS we use a novel Tandem initialisation method [46] that results in faster convergence of the iterative process, yields more stable results, and increases the segmentation accuracy.

5.9 Dimensionality Reduction for Decomposition

Although RPCA-based methods are promisingly successful in providing a good model of the background and better foreground detection in the case of moving cameras, they still suffer from the curse of dimensionality and poor scalability. As the resolution of the images and the length of the video increase, RPCA becomes progressively computationally inefficient, making them unsuitable for any practical use. In surveillance, although the resolution of the images are usually small, the length of the video is tremendously large. Computational cost of RPCA methods lies mainly in the *SVD* calculation step for modelling the background. For a large high-resolution video, the algorithm requires updating the low-rank matrix L in each iterative step, that involves performing the

expensive calculation of the κ largest singular values and vectors of a $m \times n$ matrix.

Different strategies for the dimensionality reduction for RPCA-based methods have been proposed in the literature. The bilateral random projections proposed by [71] reduce the dimension but the results tend to be highly dependent on initial random matrix used by the algorithm. In [74] the authors proposed a real-time technique based on the reduction of the number of entries in each frame, however the background is updated frame by frame and this affects the performance of foreground segmentation. Here, we propose a novel dimensionality reduction technique that calculates the background model from a sketch of the video. Recently, there has been a lot of interest on selecting the *best* or *most representative* columns from a data matrix. Qualitatively, these columns reveal the most important information hidden in the underlying matrix structure. This is similar to what principal components carry, as extracted via PCA. In sharp contrast to PCA, using actual columns of the data matrix to form a low-rank surrogate offers interpretability, making it more attractive for the problem at hand. This problem which is referred to as *Subset Selection* or *Column Subset Selection Problem* (CSSP), is a method for selecting a subset of columns from a real matrix, so that the subset represents the entire matrix well and is far from being rank deficient [14]. In background modelling, the matrices containing the video sequence can be so large that there is not enough memory to work with the whole matrix. In these cases, one needs to identify a smaller part of the matrix that represents the whole matrix well. The theoretical computer science community has come up with randomised [56], [111] and deterministic [74], [34] algorithms that use probability distributions to find the most representative columns in a matrix [9], [14].

The CSSP problem is defined as: Let $A \in \mathbb{R}^{m \times n}$ and let $c < n$ be a sampling parameter. Find c columns of A – denoted as $C \in \mathbb{R}^{m \times c}$ – that minimise

$$\|A - CC^{\dagger}A\|_F \quad or \quad \|A - CC^{\dagger}A\|_2 \tag{5.19}$$

Algorithm 3 Pseudo-code for Deterministic CSSP

Input: $A \in \mathbb{R}^{m \times n}$, κ, θ
Calculate the top κ singular vectors of A as $V_\kappa \in \mathbb{R}^{n \times \kappa}$.
for $i = 1, 2, \ldots, n$ **do**
  $\ell_i^\kappa = \|V_\kappa(i, :)\|_2^2$
end for
Without loss of generality, let ℓ_i^κ be sorted:
  $\ell_1^\kappa \geq \cdots \geq \ell_i^\kappa \geq \ell_{i+1}^\kappa \geq \cdots \geq \ell_n^\kappa$
Find index $c \in \{1, \ldots, n\}$ such that $c = \arg\min_c (\sum_{i=1}^{c} \ell_i^\kappa > \theta)$.
If $c < \kappa$, set $c = \kappa$.
Output: $\mathcal{A} \in \mathbb{R}^{n \times c}$ s.t. $A\mathcal{A}$ has the top c columns of A.

where $C^\dagger$ denotes the Moore-Penrose pseudo-inverse. We can equivalently write $C = A\mathcal{A}$, where the *sampling matrix* is $\mathcal{A} \in \mathbb{R}^{n \times c}$. State-of-the-art algorithms for CSSP utilise both deterministic and randomised techniques; we therefore consider both here. A simple but extremely successful deterministic strategy is proposed [82] which is based on sampling columns of A that correspond to the largest leverage scores ℓ_i^κ, for some $\kappa < rank(A)$. As the number of columns to be selected is not known a priori, the algorithm selects the c columns of A that correspond to the largest c leverage scores ℓ_i^κ such that their sum $\sum_{i=1}^{c} \ell_i^\kappa$ is more than an "energy" parameter θ. This ensures that the selected columns have accumulated energy at least θ. We have to carefully pick θ, our *stopping threshold*; this parameter essentially controls the quality of the approximation. We follow the theoretical recommendations of [123] for selection of θ, such that the sampling matrix $\mathcal{A}$ preserves the rank of V_κ^T in $V_\kappa^T\mathcal{A}$, i.e., choose θ such that $rank(V_\kappa^T\mathcal{A}) = \kappa$; where $V_\kappa \in \mathbb{R}^{n \times \kappa}$ contains the top κ right singular vectors of the matrix $A \in \mathbb{R}^{m \times n}$ with rank $r = rank(A) \geq \kappa$. Then, the rank-κ leverage score of the i-th column of A is defined as

$$\ell_i^\kappa = \|V_\kappa(i, :)\|_2^2, \quad i = 1, 2, \ldots, n \tag{5.20}$$

Here, $V_\kappa(i, :)$ denotes the i-th row of V_κ. Algorithm 3 shows the pseudo-code for the deterministic CSSP.

A more sophisticated method that circumvents the lack of theoretical analysis of the above deterministic algorithm, uses randomisation; although it has been proven that

Algorithm 4 Pseudo-code for Randomised CSSP

1: **Input:** $A \in \mathbb{R}^{m \times n}$, κ
2: *For a target rank $\kappa < rank(A)$, define a probability distribution over columns of A as:*
3: $$\xi_i = \frac{\ell_i^\kappa}{\kappa}, \quad i = 1, \ldots, n$$
4: *In c i.i.d. passes, sample with replacement c columns form A with probabilities given from ξ_i.*
5: **Output:** The random subset of columns $C \in \mathbb{R}^{m \times c}$.

the above sampling can be as accurate as its randomised counterparts [123]. In [34] the authors use the leverage scores to find a probability vector $\xi_i = \ell_i^\kappa/\kappa, \quad i = 1, \ldots, n$, where each i-th component is interpreted as the probability of the i-th column to be selected. Observe that $\sum_i \xi_i = 1$, since $\sum_i \ell_i^\kappa = \|V_\kappa\|_F^2 = \kappa$. An important remark that needs to be made is that the randomised algorithm above yields a matrix estimate that is "near-optimal", i.e., has error close to that of the best rank-κ approximation. Algorithm 4 shows the pseudo-code for the randomised CSSP.

Based on the presented methodology, we perform the background modelling using the output of CSSP algorithm, where a lot of redundant information is discarded, as it does not contribute to the background model, if even worse, does not contaminate it. The background model is essentially formed by the vectors of a basis of a subspace of dimension c (much smaller than the number of frames). Although in principle all columns of A are used for construction of this basis, in practice only a group of frames are determinant for the background calculation. This dimensionality reduction is of utmost importance, because as mentioned before the background calculation step is the most computationally expensive part of the RPCA-based methods. The number of floating operations (FLOPS) by iteration $min(mn^2, m^2n)$ is reduced significantly to $min(mc^2, m^2c)$. The calculated background model L is the orthonormal projection of the columns of $A - S$ onto the r-dimensional subspace that is closest to the subspace spanned by the selected columns of the matrix $A - S$. This background model is then used in the approximated RPCA framework, for foreground segmentation with our DBSS and DSPSS models.

Another interesting remark with using the CSSP strategy for background modelling

is that there would be no need for the training stage, where a clean background (without any foreground objects) is used to properly obtain a background model. Therefore, for our experiments we completely ignore the training stage (if exists for a test sequence) and test our algorithm using the *temporal region of interest* of the test sequences. In our experiments we have found out that, even if a completely clean background does not exist during the whole testing stage, the CSSP algorithm can successfully find those columns of A that can best estimate a subset that represents the entire A; in other words, it successfully selects the best columns that are extremely close to the rank-κ approximation of A with theoretical guarantees. A good background model will directly affect foreground segmentation accuracy, and our segmentation results once again confirm the efficiency and efficacy of the proposed background modelling framework.

5.10 Experiments and Analysis

We present qualitative and quantitative results for two algorithms proposed in this chapter, DBSS and DSPSS both with tandem initialisation and deterministic CSSP for background modelling. All the tests were conducted on the *temporal region of interest* of the sequences, meaning no training stage with clean background was used to obtain the background model. The algorithms are implemented in MATLAB and run on a desktop machine, using a single core on an Intel Core i7-4770 CPU and 32 GB of RAM. The average processing time on a sequence of 100 RGB frames with resolution 600×800 with image alignment and background motion estimation is about 665 seconds for DBSS and 1674 seconds for DSPSS excluding the superpixel generation step. With CSSP these times decrease accordingly to 195 seconds for DBSS and 488 seconds for DSPSS, meaning that time consumption is decreased more than 3.4 times. It is worth mentioning that the amount of time required for RPCA-based methods substantially increases with the number of frames, and one would eventually run out of memory. Hence, without CSSP, the time consumption trend is non-linear and going to explode.

Table 5-A: Description of the parameters for DBSS and DSPSS.

DBSS	λ	Regularising parameter.	$\frac{3}{\sqrt{\max(m,n)}}$
	d	Depth of each tree.	3
	m	Number of singleton groups.	64
	θ	Energy value for CSSP.	$.25 \times n$
DSPSS	λ	Regularising parameter.	$\frac{3}{\sqrt{\max(m,n)}}$
	d	Depth of each tree.	3
	$\mathcal{M}$	Number of singleton groups.	Dynamic
	k_clusters	Number of superpixels per image.	800
	c_factor	Compactness factor, controlling adherence of each superpixel region to object boundaries.	20
	θ	Energy value for CSSP.	$.25 \times n$

Four datasets are used in our experiments:

- *SABS* Brutzer *et al.* [15], a synthetic dataset.[2]
- *WallFlower* Toyama *et al.* [152].[3]
- *i2R* Li et al. [91].[4]
- *Change Detection (CDnet) 2012* Goyette *et al.* [160], an online chart which is actively updated with state-of-the-art methods in background subtraction and foreground detection.[5]

We perform extensive tests using these datasets comprised of a total of 49 videos, allowing us to compare our method to a large number of alternative methods. For all the tests the same set of parameters are used (reported in Table 5-A), unless otherwise stated.

It must be noted that the reason for selecting these datasets as opposed to more recent large-scale datasets with multi-class labels, such as ImageNet [131], MS COCO [93], CITYSCAPES [26], and PASCAL VOC [51] is that our chosen datasets include video sequences, where the temporal redundancy between the frames of the video sequences

[2]Online at `http://www.vis.uni-stuttgart.de/index.php?id=sabs`.
[3]`http://research.microsoft.com/en-us/um/people/jckrumm/WallFlower/TestImages.htm`.
[4]`http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html`.
[5]Dataset and chart online at `http://www.changedetection.net`.

Table 5-B: Relative error and time cost of CSSP for low-rank approximation of four 100 frame test sequences.

size(A)	rank (L)	Energy θ	rel. error ($\|L-\hat{L}\|_F/\|L\|_F$) (10^{-3})	distance between largest singular values $\|\sigma_1^L - \sigma_1^{\hat{L}}\|$ (10^{-6})	time (seconds)
128×160	1	.01/.05/.10/.25/.5	.364/.194/.119/.083/.073	1133.00/183.96/14.13/38.64/192.86	8.38/8.51/8.51/8.73/9.37
	5	.10/.25/.5	.842/.382/.432	240.06/336.84/857.67	8.12/8.32/8.61
	10	.25/.5	.367/.447	759.257/356.77	8.62/9.17
	25	.5	.182	310.69	12.73
144×176	1	.01/.05/.10/.25/.5	.658/.281/.156/.110/.097	6739.10/746.69/681.16/56.09/102.24	10.58/10.54/10.60/10.94/11.43
	5	.10/.25/.5	.540/.575/.343	8.89/294.42/28.05	10.12/10.29/10.78
	10	.25/.5	.335/.316	2515.90/1381.5	11.74/12.89
	25	.5	.279	1084.4	15.39
240×360	1	.01/.05/.10/.25/.5	.065/.045/.034/.021/.016	14.81/71.05/25.11/20.29/11.52	28.91/29.00/29.16/30.19/31.62
	5	.10/.25/.5	.152/.152/.117	379.55/52.10/45.63	31.65/32.80/34.74
	10	.25/.5	.124/.117	33.93/28.64	36.59/40.03
600×800	1	.01/.05/.10/.25/.5	.339/.324/.152/.123/.086	3828.20/1678.9/429.59/19.90/63.27	220.92/186.73/178.85/186.41/193.30
	5	.10/.25/.5	.795/.735/.423	1223.3/128.78/601.65	217.12/195.87/199.83
	10	.25/.5	.428/.337	569.83/252.05	216.63/237.82
	25	.5	.225	351.44	304.80

could be exploited to estimate a low-rank component for the input data. The multi-class datasets above include only a single image per sample, although multiple instances of the same class might exist. Therefore, it is virtually impossible to test our method on such datasets.

5.10.1 Efficacy of CSSP

Table 5-B shows the relative error and time cost of CSSP for low-rank approximation of four 100-frame test sequences with different sizes. We have tested several rank approximations with different energy values for θ, and report the relative error between CSSP denoted as $\hat{L}$ and low-rank approximation denoted as L in terms of the Frobenius norm and distance between the largest singular values. The time cost for each approximation is displayed at the end of each row.

We also tested the CSSP for the four benchmark datasets described above. Figure 5.6 shows the PSNR values obtained by using 20 values of θ linearly distributed in range $[.05, 1]$. According to this for all the our tests using 25% of the columns of A guarantees a very accurate model of the background, while a larger θ will not always result in significant increase in PSNR. An important observation here which demonstrates the advantage of using CSSP, is that, as we introduce more frames to the background (i.e., we use higher θ) we risk contaminating the background model by more foreground

Table 5-C: Summary of selected videos from our test datasets used in CSSP experiments.

Dataset	Video	$m \times n$
i2R	WaterSurface	20480×100
Wallflower	hall	25344×100
CDnet 2012	pedestrians	86400×100
SABS	basic	480000×100

information; this is seen the fluctuations in Figure 5.6-(b), (c), (e), (g), (h). That means an optimal θ is rather one that is smaller, that will select the most representative frames for the background of a sequence.

Figures 5.7 demonstrates total time consumption for processing a 100 frame video in each of the datasets with varying θ in comparison with original low-rank modelling. Again our choice of θ lies in the elbow of these plots and provides time-saving guarantees.

In this subsection we investigate the empirical performance of the CSSP on real datasets. Our experiments are not meant to be exhaustive; however, they provide clear evidence that deterministic leverage score sampling in real world matrices is particularly effective. It has been shown in [123] that deterministic CSSP can obtain a moderately small relative error $\frac{\|A-CC^{\dagger}A\|_2^2}{\|A-A_\kappa\|_2^2}$ with significantly smaller number of selected columns c in both real and synthetic matrices. Here, we analyse the relative error of approximation of A by $CC^{\dagger}A$, with respect to its rank-κ approximation by A_κ in the same manner as [123] on sample 100-frame videos from our 4 datasets. We have chosen representative videos from each dataset shown in table 5-C. Figure 5.8 shows the relative error ratio achieved by Algorithm 3 as a function of the energy value θ where $c = \theta \times n$. The small relative errors obtained in these examples indicates that the selected value for the energy parameter $\theta = .25$ suffices for an approximation as good as that of the best rank-κ approximation obtained by SVD.

5.10.2 CDnet 2012 dataset

The change detection dataset [160] is the largest dataset in our evaluation, and includes a dense ground truth that is provided for all frames past some initial training period, and in some cases a *Region of Interest* (ROI) for temporal and spatial evaluation of challenging parts of the videos. It also limits parameter tuning, such that a single parameter must be used for all the 31 videos. Video resolution is not great however, with many videos appearing as if they have been post-processed, often with a low quality de-interlacing algorithm that creates ghosts. The dataset is comprised of six categories, 31 real-world videos (including thermal sequences), totaling over 80,000 frames, to include diverse motion and change detection challenges.

- *Basline.* A basic set of videos but not trivial to process. Some videos with background motion, others with isolated shadows, abandoned objects, slow moving foreground, and saturated colour for one of the videos.

- *Camera jitter.* The camera is not properly mounted for these videos, and the resulting jitter magnitude varies form one video to another. This does not reduce the quality of our output, as our algorithm manages to fully compensate for background motion, even during the worst shakes, thanks to the pre-alignment step and motion parameter estimation simultaneously with decomposition; hence both our DBSS and DSPSS algorithms achieve top rank.

- *Dynamic background.* These videos have strong parasitic background motion, such as shimmering water, a fountain, and a tree swaying by the wind. As demonstrated in the results section, our algorithm excels at such input, outperforming competitors significantly.

- *Intermittent object motion.* These videos are aimed at causing "ghosting" artifacts in the detected motion, i.e., objects move, then stop for a short while, only to start moving again afterwards. This category is intended for testing how an

algorithm adapts to background changes. We advantage at this category thanks to the tandem initialisation to remove the ghosting problem, and the robust low-rank approximation of the background, that can learn multiple modes for the background of a sequence.

- *Shadow.* These videos have strong to faint shadows, and test the ability of an algorithm to ignore them. As such our algorithm is capable of handling soft shadows, and if the shadow is cast over several plateaus (i.e., broken on an escalator or a sidewalk and the road) or is fairly narrow, the structured-sparsity will do a fair job at eliminating these. Apart from these our algorithm does not have any specific code to handle shadows, whilst we do not. Interestingly our result remains the top approach despite its limited capabilities.

- *Thermal.* An unusual set of videos for background subtraction but very useful for industrial purposes or surveillance where far-infrared cameras are used primarily. These videos contain an unsurprising amount of noise expected from this input, typical thermal artifacts such as heat stamps (e.g, bright spots left on a seat after a person gets up and leaves), heat reflection on floors and windows, and camouflage effects when a moving object has the same temperature as the surrounding regions. Both DBSS and DSPSS suffer from these challenges and achieve average results in this section.

The results for these sequences can be seen in Figures 5.11, 5.12, 5.13, 5.14, and 5.15. Quantitative results can be found in Table 5-D.[6] For each category we compare our DBSS and DSPSS algorithms with the top performing methods which have submitted results for that category for the reason of space limit (readers are referred to [160]) and its website for complete list of references and the corresponding performance figures). In addition to this list, we have included the DP-GMM [69] and five RPCA-based methods PCP[7] [179], DECOLOR [178], and very recent 2-pass RPCA [58]. For LSD-GSRPCA

[6]The Table is accurate as of January 2018–all results reported can be found at `http://changedetection.net`.

[7]For PCP a thresholding step is required to produce the final foreground masks, as many entries in S

[101] and SPGFL [79] only a fraction of the results were reported in their papers, therefore they are included where results are reported. For PCP we use our pre-alignment step for the *camera jitter* sequences and as such we denote it as PCP+Alignment. The online version of CDnet combines many different scoring mechanisms, and then combines them in a non-linear rank based system. Instead, we present the F-measure scores only, as it is the most used metric. The F-measure is defined as the harmonic mean of the recall and precision:

$$\text{recall} = \frac{tp}{tp + fn}, \tag{5.21}$$

$$\text{precision} = \frac{tp}{tp + fp}, \tag{5.22}$$

$$\text{F-measure} = 2\frac{\text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}}, \tag{5.23}$$

where fp is the number of false positives, tn the number of true negatives, etc. Overall, we win on average for the CDnet dataset both for DBSS and DSPSS. This is because our model can handle backgrounds that are complex and dynamic. This ability, in combination with the tree-structured sparsity inducing mechanisms allows it to effectively segment genuine well-outlined foreground regions.

5.10.3 SABS dataset

The SABS dataset [15] presents synthetic image sequences divided into nine categories. Despite being synthetic it simulates, fairly accurately, various challenges that are not present in the CDnet dataset (e.g., sudden illumination change, high noise in dim conditions, and camouflage). It has the advantage of having ground truth for all frames in

may contain vanishingly small values. To obtain a threshold, first the likely outlier locations is identified. Those pixels whose corresponding entries in S have magnitudes less than half of the maximum entries in S are regarded as background. Next the difference between A and L at those tentatively identified background locations are obtained to estimate the expected level of noise. Finally, the threshold is set to the mean of the difference values plus three standard deviations of those difference values and is applied to S.

Table 5-D: *CDnet 2012** [160] dataset: F-measure results for all the categories for the most competitive methods.

method	Baseline	Camera Jitter	Dynamic Background	Intermittent Motion	Shadow	Thermal	*mean*
LSD-GSRPCA [101]	.7173 (19)	-	-	-	-	-	-
SPGFL [79]	.9469 (3)	-	.8519 (5)	.6988 (7)	-	.8156 (5)	-
SGMM [49]	.8594 (18)	.7251 (13)	.6380 (18)	.5397 (16)	.7944 (14)	.6481 (18)	.7008 (17)
ViBe+ [4]	.8715 (17)	.7538 (10)	.7197 (11)	.5093 (18)	.8153 (9)	.6646 (17)	.7224 (16)
SC-SOBS [110]	.9333 (7)	.7051 (16)	.6686 (17)	.5918 (12)	.7786 (17)	.6923 (16)	.7283 (15)
PCP+Alignment [179]	.9109 (16)	.7218 (15)	.6941 (14)	.5371 (17)	.7885 (16)	.7192 (12)	.7286 (14)
PSP-MRF [134]	.9289 (8)	.7502 (11)	.6960 (13)	.5645 (14)	.7907 (15)	.6932 (15)	.7372 (13)
PBAS [73]	.9242 (13)	.7220 (14)	.6829 (15)	.5745 (13)	.8597 (6)	.7556 (9)	.7532 (12)
DECOLOR [178]	.9215 (15)	.7776 (9)	.7084 (12)	.5945 (11)	.8317 (7)	.7081 (14)	.7570 (11)
SGMM-SOD [50]	.9223 (14)	.6988 (17)	.6826 (16)	.6957 (8)	.8613 (5)	.7081 (13)	.7624 (10)
DP-GMM [69]	.9286 (11)	.7477 (12)	.8137 (7)	.5418 (15)	.8127 (10)	.8134 (6)	.7763 (9)
2-pass RPCA [58]	.9281 (12)	.8152 (6)	.7818 (10)	.6826 (9)	.8063 (13)	.7597 (8)	.7956 (8)
MBS V0 [132]	.9287 (10)	.8367 (5)	.7904 (9)	.7092 (6)	.8063 (12)	.8115 (7)	.8092 (7)
MBS [133]	.9287 (9)	.8367 (4)	.7915 (8)	.7568 (5)	.8262 (8)	.8194 (3)	.8217 (6)
SuBSENSE [141]	.9500 (2)	.8150 (7)	.8180 (6)	.6570 (10)	.8990 (3)	.8170 (4)	.8260 (5)
PAWCS [140]	.9397 (6)	.8137 (8)	.8938 (4)	.7764 (4)	.8710 (4)	.8324 (2)	.8545 (4)
CDet [2]	.9458 (4)	.8367 (3)	.8991 (3)	**.8039 (1)**	.8122 (11)	**.8337 (1)**	.8552 (3)
DBSS	.9430 (5)	**.8804 (1)**	.9005 (2)	.7837 (3)	.9107 (2)	.7195 (11)	.8563 (2)
DSPSS	**.9664 (1)**	.8662 (2)	**.9057 (1)**	.7870 (2)	**.9177 (1)**	.7328 (10)	**.8626 (1)**

*Table accurate as of January 2018, with results from CDnet `http://changedetection.net/`. The online chart keeps updating.

the nine categories:

- *Basic.* A baseline test for general performance.
- *Dynamic Background.* A crop of the area of analysis to the area of waving tree and changing traffic light.
- *Bootstrapping.* No training phase, thus subtraction starts after the first frame.
- *Darkening.* Gradual illumination change to simulate sun setting.
- *Light Switch.* Once-off changes by switching the lights of a shop off and on again.
- *Noisy Night.* Basic sequence at night, with increased sensor noise accounting for high gain level and low background and foreground contrast resulting in more camouflage.
- *Camouflage.* Persons and cars coloured similar to the background, so they are hard to distinguish.
- *No Camouflage.* Same as camouflage, but with easy to see colours for comparison.
- *Video Compression (H264-40kbps).* Heavily compressed videos, to generate typical compression artifacts.

As can be seen in the results in Table 5-E, our DSPSS algorithm takes the first place in all the scenarios except for *light switch*. Our background model slowly adapts to changes in the scene, and this takes its toll on our method in this challenge. If the rank of the low-rank component is too high, it would compensate for these changes, but will probably quickly absorb many slow-moving foreground regions into the background. On the other hand, if the rank is too low, it will not adapt to modality changes in the background well. Hence, this is a trade-off situation for our method. The DSPSS wins on average, and DBSS stands 3rd after DP-GMM. It must be noted that the other algorithms have had their post-processing removed, therefore, it is a fairer comparison

Table 5-E: *SABS* [15] dataset: F-measure results for nine challenges; only the most competitive algorithms were included.

method	basic	dynamic background	bootstrap	darkening	light switch	noisy night	camouflage	no camouflage	H.264, 40Kbps	*mean*
Stauffer [142]	.800 (4)	.704 (6)	.642 (6)	.404 (8)	.217 (7)	.194 (7)	.802 (5)	.826 (5)	.761 (7)	.594 (8)
Maddalena [109]	.766 (6)	.715 (4)	.495 (8)	.663 (6)	.213 (8)	.263 (6)	.793 (6)	.811 (6)	.772 (6)	.610 (7)
Li 1 [90]	.766 (6)	.641 (7)	.678 (5)	.704 (4)	.316 (4)	.047 (8)	.768 (7)	.803 (7)	.773 (5)	.611 (6)
Barnich [5]	.761 (7)	.711 (5)	.685 (4)	.678 (5)	.268 (6)	.271 (5)	.741 (8)	.799 (8)	.774 (4)	.632 (5)
Zivkovic [181]	.768 (5)	.704 (6)	.632 (7)	.620 (7)	.300 (5)	.321 (4)	.820 (4)	.829 (4)	.748 (8)	.638 (4)
DP-GMM, with post [69]	.853 (2)	.853 (2)	.796 (3)	.861 (2)	**.603 (1)**	.788 (2)	.864 (3)	.867 (3)	.827 (2)	.812 (2)
DBSS	.823 (3)	.701 (3)	.798 (2)	.850 (3)	.496 (3)	.715 (3)	.878 (2)	.890 (2)	.806 (3)	.784 (3)
DSPSS	**.867 (1)**	**.871 (1)**	**.822 (1)**	**.907 (1)**	.570 (2)	**.897 (1)**	**.894 (1)**	**.913 (1)**	**.841 (1)**	**.842 (1)**

for our method; however, we included DP-GMM with post-processing.

5.10.4 i2R dataset

The i2R dataset [91] is similar to WallFlower [152] dataset. The testing procedure is similar to before. The test sequences are much harder due to low quality, high noise, text overlays, and camera jitter. There are fewer algorithms that have reported results on this dataset. We have reported for DBSS and DSPSS results with and without parameter tuning per problem, since some methods in comparison have used tuning and some have not. The qualitative results can be seen in Figure 5.10 and F-measure results can be seen in Table 5-F. We achieve top performance again in all categories except for *lb* sequence, that contains abrupt lighting changes, which is compensated for slowly by our background model. Our DBSS algorithm without parameter tuning in this Table achieves a modest 5th place as a result of suffering during *lb*, but the DSPSS remains at the top place regardless.

5.11 Summary

In this chapter, we have presented a new background subtraction method and validated its efficacy and effectiveness with extensive testing. The method is based on an existing model, namely RPCA, but with new sparsity-inducing norms and group-structured sparsity constraints. Whilst our simple DBSS model produces crisp and well-defined genuine

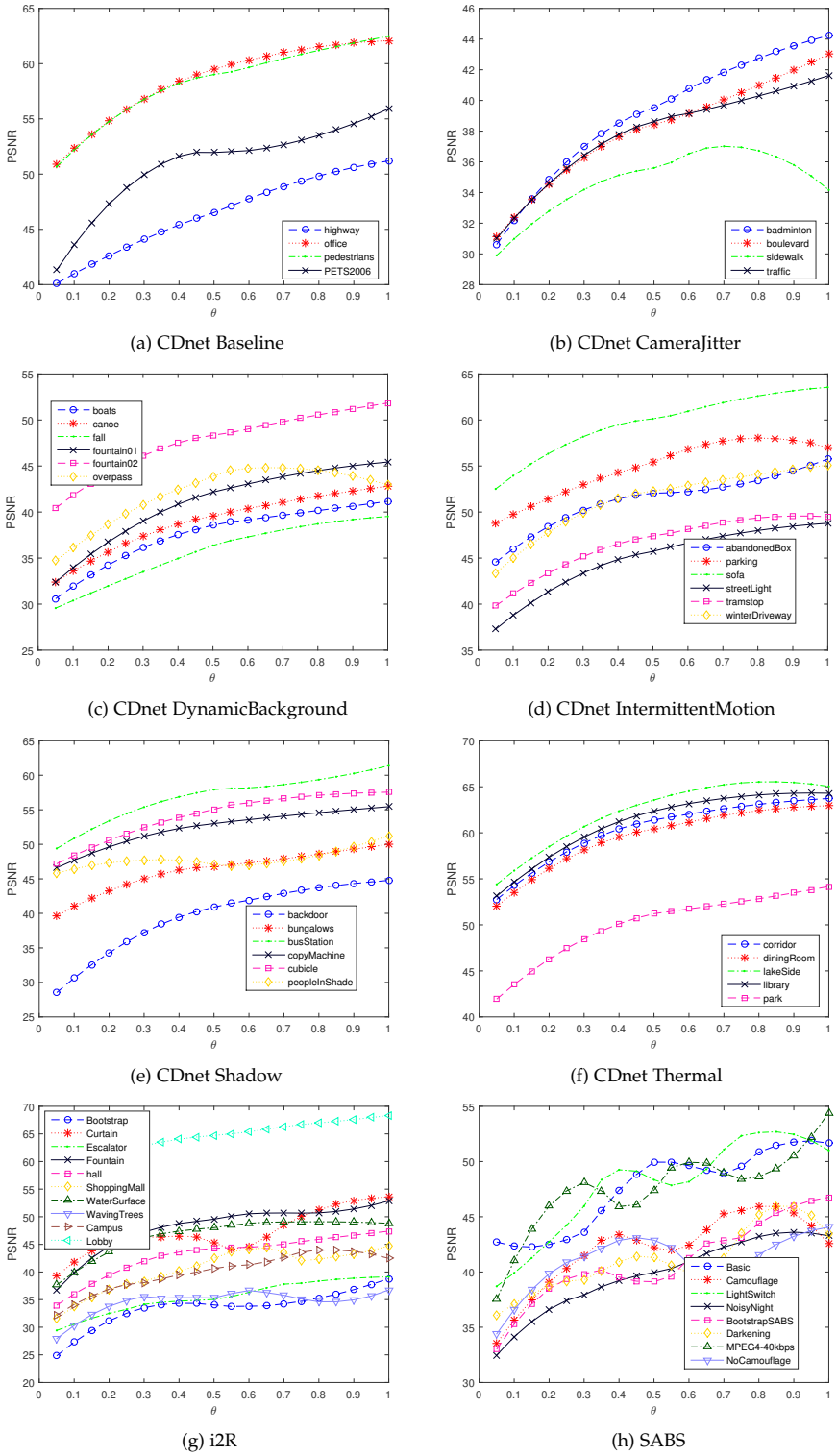
Table 5-F: *i2R* [91] and *WallFlower* [152] dataset F-measure results. We report DBSS* and DSPSS* without parameter tuning, although the dataset allows this.

method	cam	ft	ws	mc	lb	sm	ap	br	ss	*mean*
Li 2 [92]	.1596 (11)	.0999 (14)	.0667 (14)	.1841 (14)	.1554 (14)	.5209 (14)	.1135 (14)	.3079 (14)	.1294 (14)	.1930 (14)
SemiSoftGoDec [176]	.0903 (12)	.2574 (12)	.4473 (13)	.4344 (13)	.3602 (13)	.6554 (11)	.5713 (10)	.3561 (13)	.2751 (12)	.3830 (13)
Stauffer [142]	.7570 (6)	.6854 (9)	.7948 (10)	.7580 (11)	.6519 (8)	.5363 (13)	.3335 (13)	.3838 (12)	.1388 (13)	.4842 (12)
Culibrk [28]	.5256 (8)	.4636 (11)	.7540 (11)	.7368 (12)	.6276 (11)	.5696 (12)	.3923 (12)	.4779 (11)	.4928 (11)	.5600 (11)
DECOLOR [178]	.3416 (10)	.2075 (13)	.9022 (8)	.8700 (6)	.646 (10)	.6822 (8)	.8169 (4)	.6589 (7)	.7480 (6)	.6525 (10)
Maddalena [109]	.6960 (7)	.6554 (10)	.8247 (9)	.8178 (10)	.6489 (9)	.6677 (10)	.5943 (8)	.6019 (9)	.5770 (9)	.6760 (9)
DP-GMM [69]	.7876 (4)	.7424 (8)	.9298 (5)	.8411 (8)	.6665 (7)	.6733 (9)	.5675 (11)	.6496 (8)	.5522 (10)	.7122 (8)
PCP [179]	.5226 (9)	.8650 (5)	.6082 (12)	.9014 (5)	.7245 (6)	.7785 (6)	.5879 (9)	.8322 (6)	.7374 (7)	.7286 (7)
LSD-GSRPCA [101]	.7613 (6)	.8371 (6)	.9050 (7)	.8357 (9)	.7313 (5)	.7362 (7)	.7222 (7)	.5842 (10)	.7214 (8)	.7594 (6)
SPGFL [79]	.8574 (4)	.9322 (2)	**.9856 (1)**	**.9744 (1)**	**.8840 (1)**	.8265 (4)	.7739 (5)	.8394 (5)	.8029 (5)	.8751 (4)
DBSS*	.8173 (5)	.7842 (7)	.9282 (6)	.8565 (7)	.5838 (12)	.8071 (5)	.7379 (6)	.8645 (4)	.8586 (4)	.8042 (5)
DBSS, tuned	.9277 (2)	.8808 (4)	.9535 (4)	.9093 (4)	.7563 (4)	.8950 (2)	.8343 (3)	.9196 (2)	.9377 (2)	.8904 (2)
DSPSS*	.8993 (3)	.9105 (3)	.9674 (3)	.9228 (2)	.7680 (3)	.8499 (3)	.8593 (2)	.8922 (3)	.9163 (3)	.8873 (3)
DSPSS, tuned	**.9610 (1)**	**.9575 (1)**	.9719 (2)	.9093 (3)	.8725 (2)	**.9156 (1)**	**.9098 (1)**	**.9440 (1)**	**.9561 (1)**	**.9331 (1)**

foreground segmentation, our more elaborate DSPSS model surpasses its performance by taking advantage of the natural shape and structure of objects in the scene. Both our sparsity models dynamically evolve to best describe genuine foreground objects in the scene, which gives them a significant advantage when it comes to handling dynamic backgrounds, or foreground aperture. To make the problem computationally scalable we proposed using deterministic and randomised CSSP for low-rank matrix estimation and analysed its efficacy rigorously. Moreover, a novel tandem initialisation method is proposed to speed up convergence and remove ghosting effects persisting in RPCA-based methods. Specifically, our model is able to learn a robust background model that can change over time, to cope with a variety of scene changes, in comparison with the existing more heuristic RPCA-based methods. It proves itself to have excellent performance in dealing with heavy noise, thanks to the approximated RPCA model where the residual error (noise) is discarded into a third matrix in the decomposition. In addition, estimation of background motion induced by a jittering or moving camera is performed simultaneously with low-rank approximation, that results in excellent performance in videos with large camera-induced motion.

A number of improvements for our model can be considered. Our model is yet another batch method, as the frames need to be stored for obtaining a background model; although we alleviated this limitation to some extent by the CSSP, further optimisation is required to achieve real-time performances. This could include a learning stage followed

by incremental updates as the frames arrive. Spatio-temporal constraints are also another area of attraction for our method. Sudden illumination changes are slowly adapted to by the background model, and hence it fails to handle some indoor lighting changes. Furthermore, a more sophisticated model should be able to handle shadows, that are not interesting for later processing. Solutions to these problems could be adapted to our method.



(a) CDnet Baseline (b) CDnet CameraJitter

(c) CDnet DynamicBackground (d) CDnet IntermittentMotion

(e) CDnet Shadow (f) CDnet Thermal

(g) i2R (h) SABS

Figure 5.6: PSNR-θ plot of modelled background by CSSP vs. low-rank modelling for CDnet [160], i2R [91], and SABS [15] datasets. With energy value $\theta = .25$ the optimality of the quality of the modelled background is ensured.

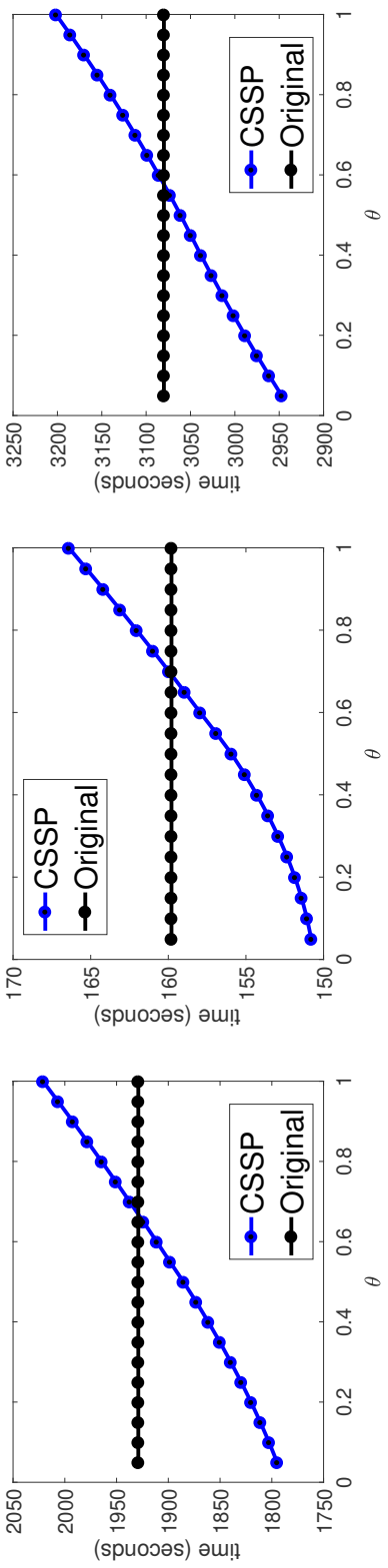


Figure 5.7: Total time consumption for processing a 100 frame sequence in our datasets. Left: CDnet [160]. Middle: i2R [91]. Right: SABS [15].

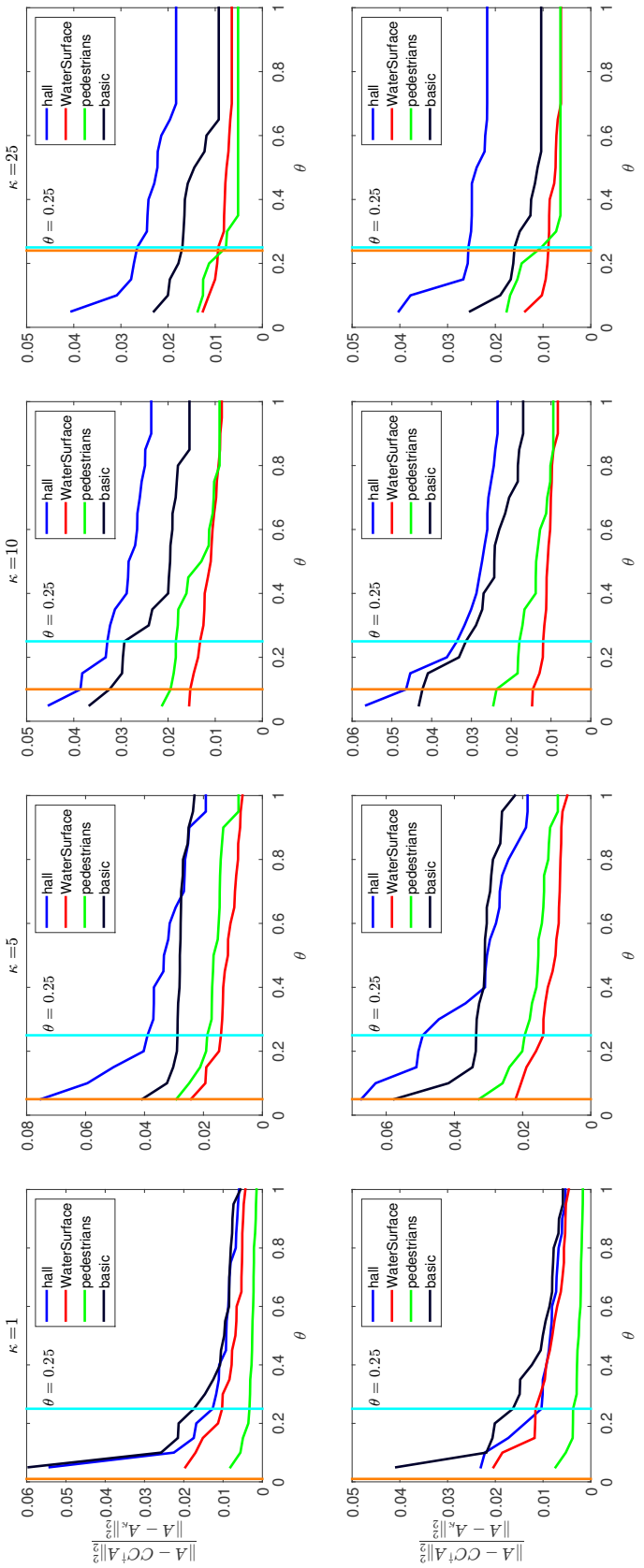


Figure 5.8: Relative error vs. θ: plot of CSSP vs. low-rank modelling. Top row: DBSS model; bottom row: DSPSS model. With energy value $\theta = .25$ the optimality of the quality of the modelled background is ensured. Here, we plot the curves for the relative error ratio $\|A - CC^\dagger A\|_2^2 / \|A - A_\kappa\|_2^2$ achieved by algorithm 3 applied to our DBSS and DSPSS models as a function of the energy value θ with $c = \theta \times n$. The leftmost vertical orange line corresponds to the point where $\kappa = c$. When $c < \kappa$ the output error is larger but negligible in all cases. The rightmost vertical cyan line indicates the point where the c sampled columns offer as good an approximation as that of the best rank-κ matrix A_κ in our experimental data.

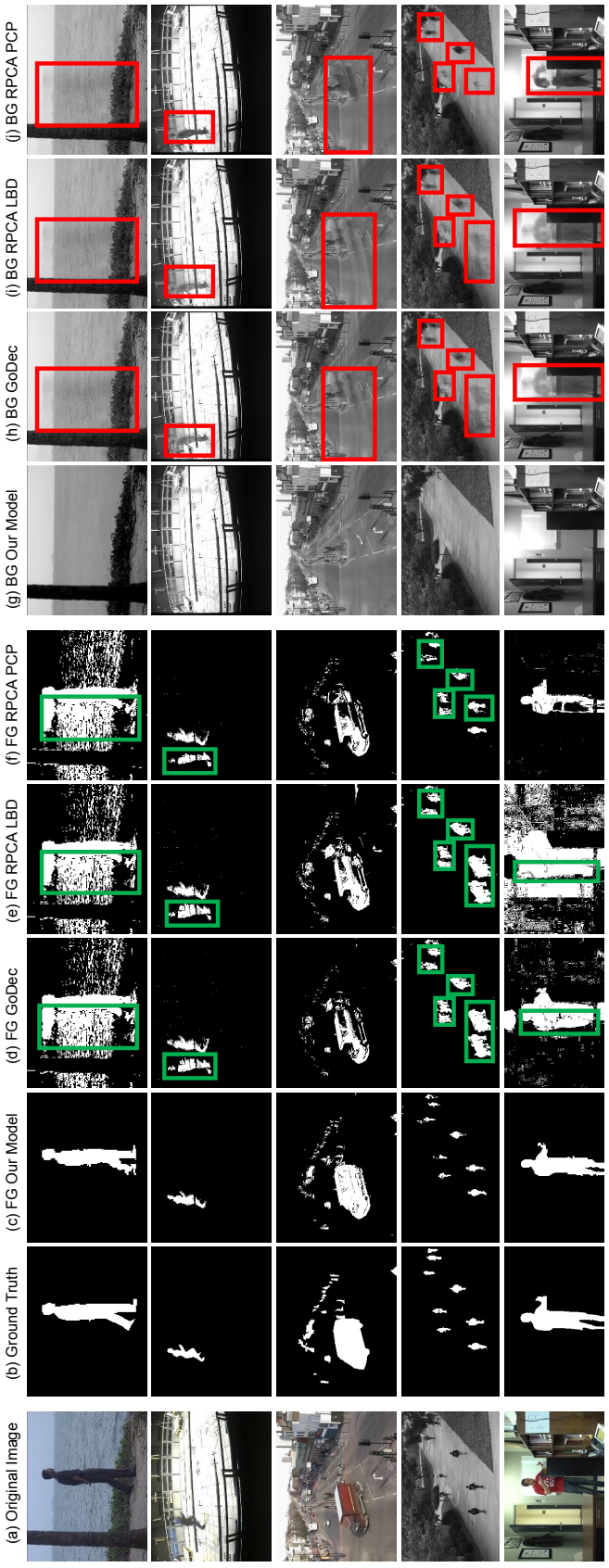


Figure 5.9: *Ghosting effects* that persist in RPCA-based methods [176], [66], [179]. A contaminated background model in red regions affects the foreground segmentation in green regions. Our tandem model is capable of eliminating these artifacts.

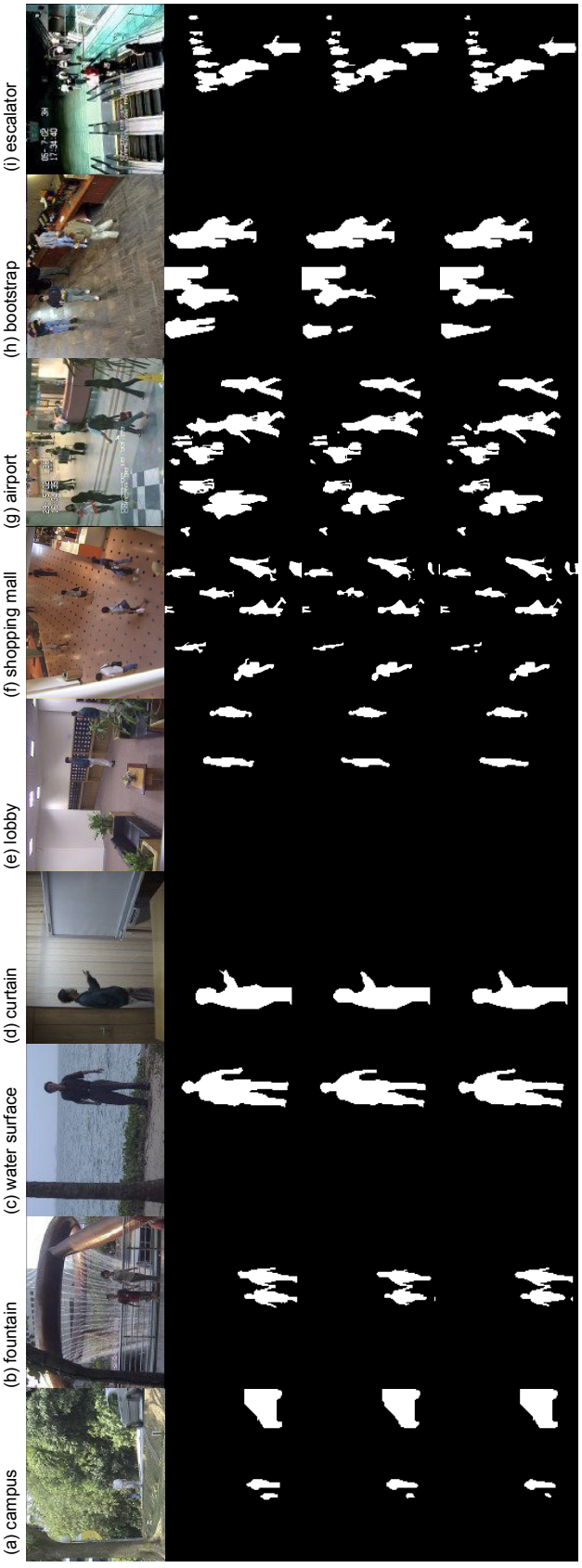


Figure 5.10: *i2R* [91] results: top row is the original image, second row is the ground truth, the third row is DBSS results, and the last row is DSPSS output. We used the same frames as [79], [101], [165], [69], [28], and [109], for qualitative comparison.

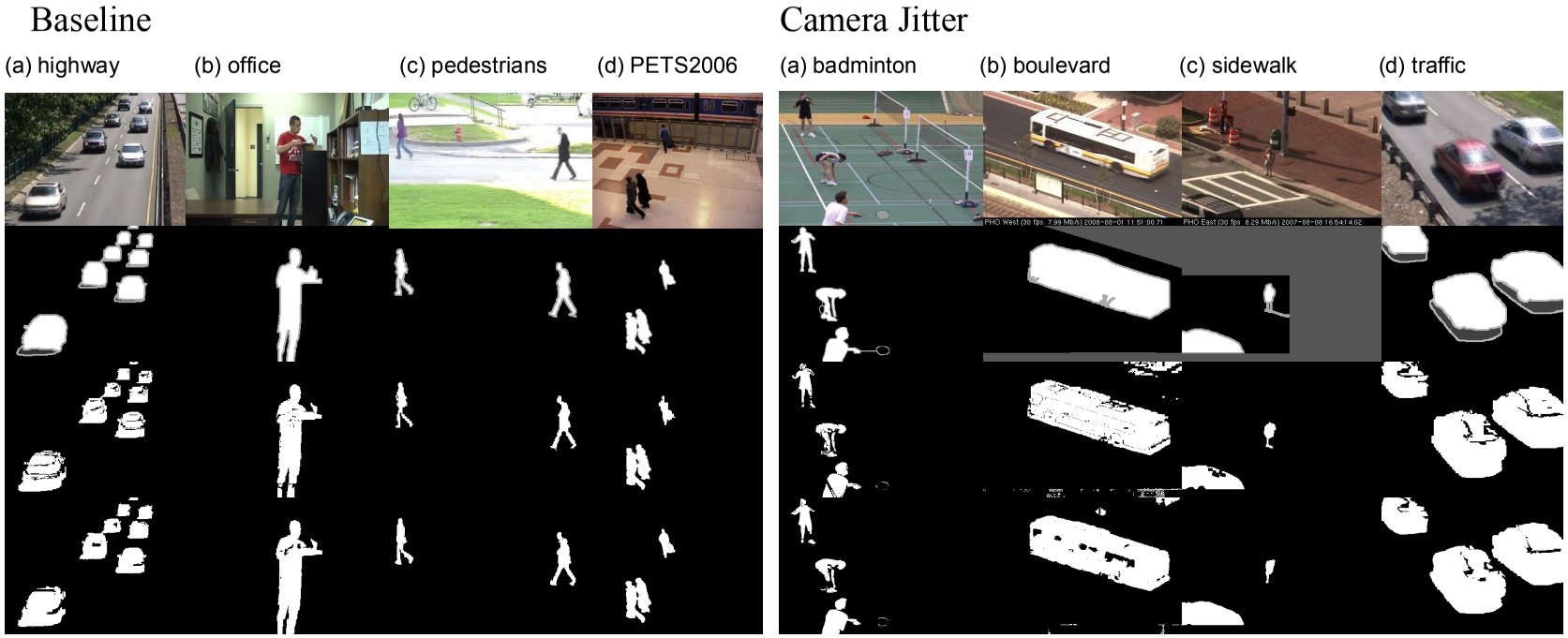


Figure 5.11: *CDnet* [160] Baseline and Camera Jitter results: identical layout to Figure 5.10 [91] with multiple rows. The ground truth includes is marked with various shades of gray – dark gray to indicate shadows, mid gray for ignored regions for evaluation, and light gray for areas ignored per frame, usually the outline of objects where foreground/background assignment is ambiguous.

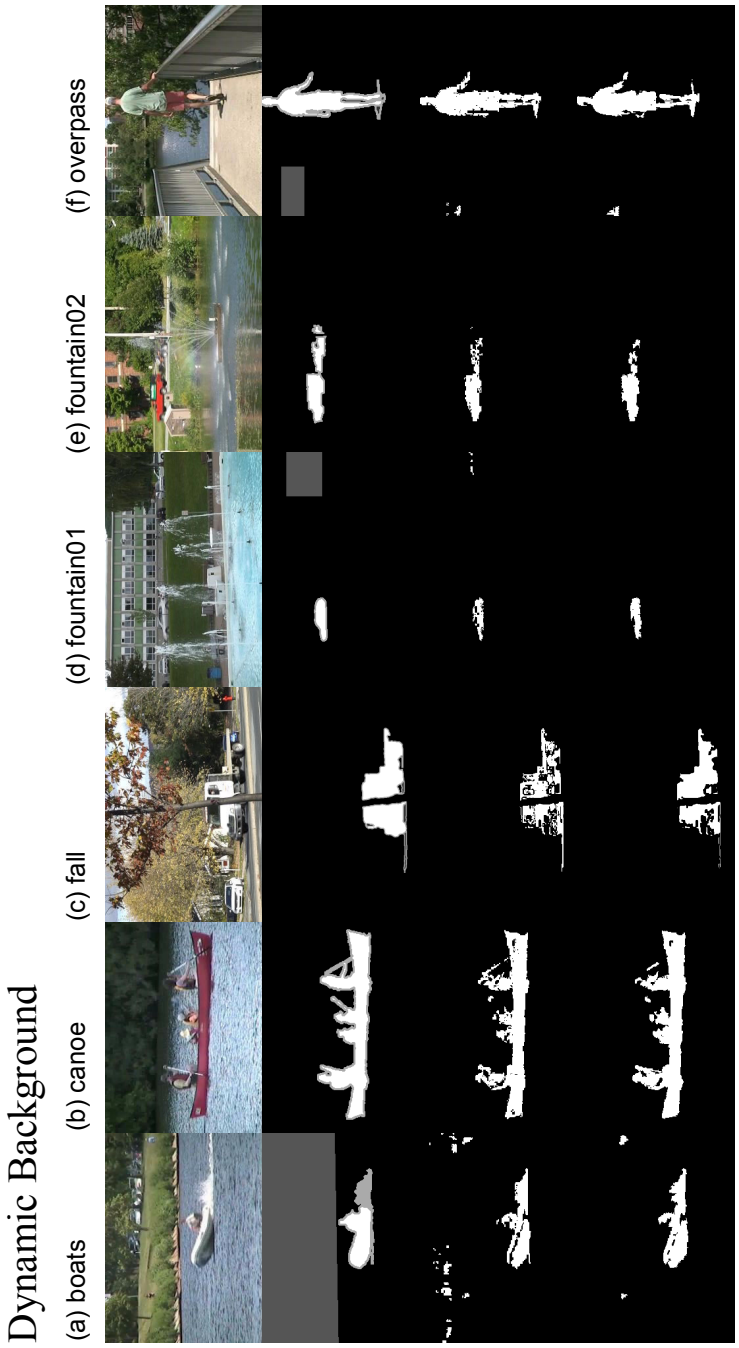


Figure 5.12: *CDnet* [160] Dynamic Background results: layout same as Figure 5.11.

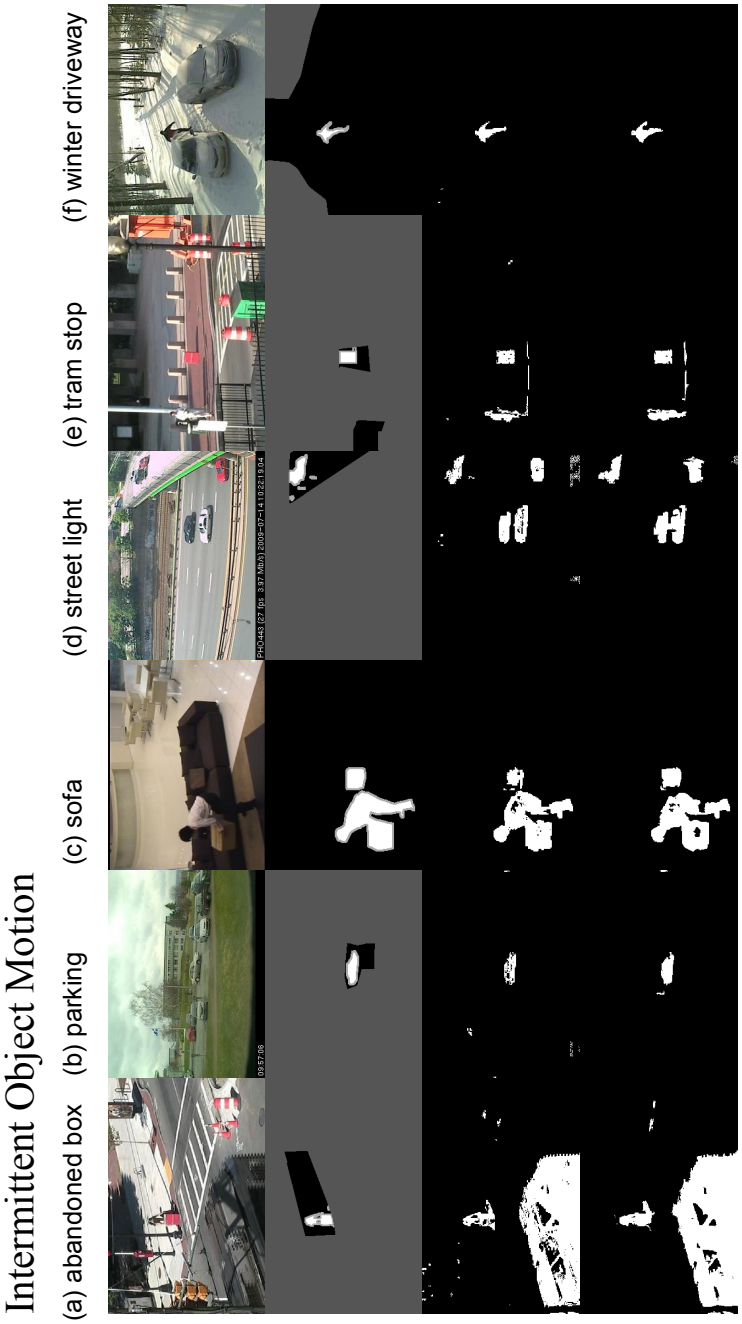


Figure 5.13: *CDnet* [160] Intermittent Object Motion results: layout same as Figure 5.11.

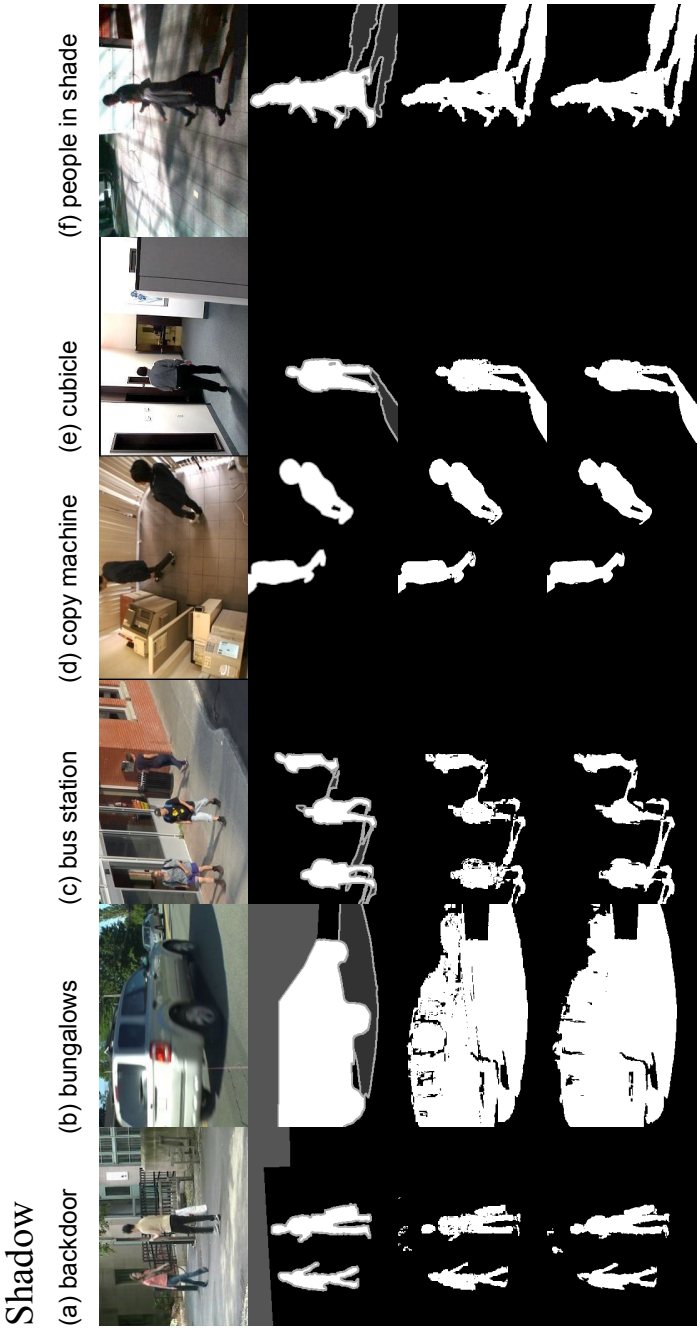


Figure 5.14: *CDnet* [160] Shadow results: layout same as Figure 5.11.

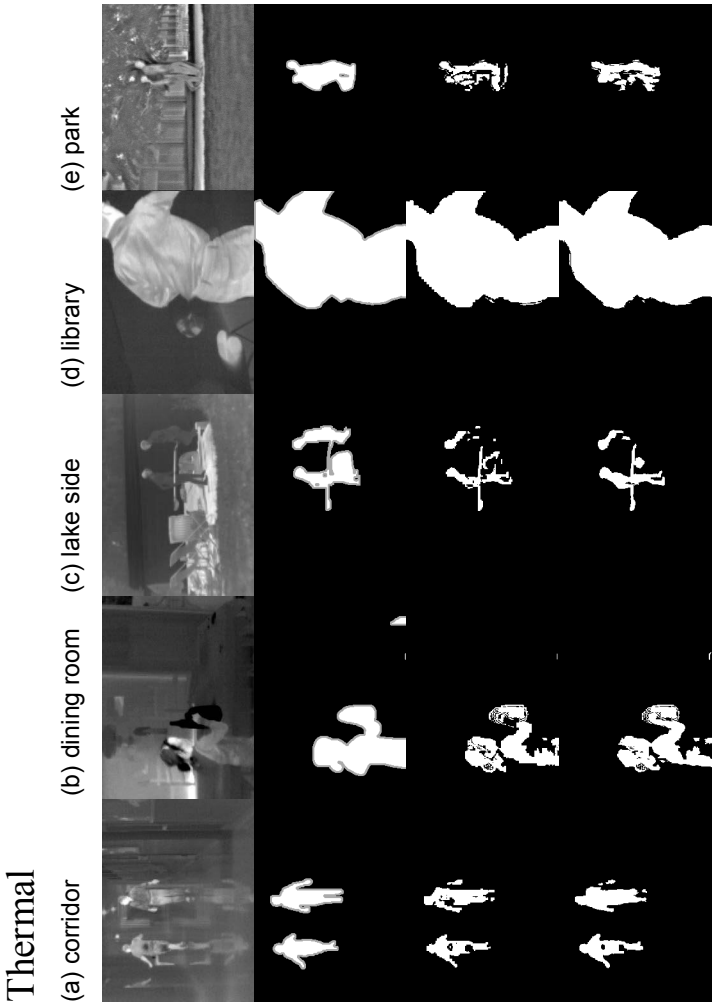


Figure 5.15: *CDnet* [160] Thermal results: layout same as Figure 5.11.

Chapter 6

Motion Subspace Clustering

In this chapter we address the problem of motion subspace clustering and segmentation. Given a set of data samples approximately drawn from a union of multiple subspaces, our goal is to cluster the samples into respective subspaces, and also remove possible outliers. We propose a novel Approximated Robust PCA Clustering (ARPCAC) method, that seeks the lowest rank representation among all the candidates that can represent the samples drawn from camera-induced motion. The proposed method involves extracting the point trajectories only induced by object motion, from the pool of all motions with our ARPCAC method, and then projecting them onto a 5-dimensional space, using PowerFactorisation. We apply our algorithm to the problem of segmenting multiple motions in video and furthermore, we extend our work to the problem of face clustering. Conducted experiments show that our approach significantly outperforms state-of-the-art methods. The findings of this chapter are published in [48].

6.1 Introduction

In pattern analysis and signal processing, an underlying intent is that the data often contains some type of *structure* that enables intelligent representation and processing.

So one usually needs a parametric model to characterise a given set of data. The well-known (linear) *subspaces* are possibly the most common choice, mainly because they are easy to compute and often effective in real applications. Several types of visual data, such as motion, face, and texture, have been known to be well characterised by subspaces. More recently, there has been an increasing interest on the geometrical and statistical models for the understanding of *dynamic* scenes, in which both the camera and multiple objects move.

The subspace methods have been gaining much attention in the recent years. For example the widely used Principal Component Analysis (PCA) method and the recently established matrix completion and recovery methods are essentially based on the hypothesis that the data is approximately drawn from a low-rank subspace. However a given dataset can seldom be well described by a *single* subspace. A more reasonable model is to consider data as lying near *several* subspaces; namely, the data is considered as samples approximately drawn from a mixture of several low-rank subspaces. The generality and importance of subspaces naturally lead to challenging problem of subspace segmentation (or clustering), whose goal is to segment (cluster or group) data into clusters with each cluster corresponding to a subspace. Subspace segmentation is an important data clustering problem that arises in numerous research areas, including computer vision (e.g., image segmentation, motion segmentation, and temporal video segmentation as illustrated in Figure 6.1), and image processing (such as image representation and compression). When the data is clean, i.e., the samples are strictly drawn from the subspaces, several existing methods (e.g., [27, 38, 97]) are able to exactly solve the subspace segmentation problem. So, the main challenge of subspace segmentation is to handle the *errors* (e.g., noise and corruptions) that possibly exist in data, i.e., to handle the data that may not strictly follow subspace structures. With this outlook, we study the following *robust subspace clustering* problem: Given a set of data samples approximately drawn from a union of linear subspaces, correct the possible errors and segment all samples into their respective subspaces, and simultaneously reveal each subspace's independent motion. By

Figure 6.1: Top: Motion segmentation. Given features points on multiple rigidly moving objects tracked in multiple frames of a video (top), the goal is to separate the feature trajectories according to the moving objects (bottom). Bottom: Face clustering. Given face images of multiple subjects (top), the goal is to find images that belong to the same subject (bottom). The same frames as [38] are shown for comparison.

independent motion we mean the camera-induced motion has been subtracted from all the motions in the scene, where the motion trajectory of an object can be revealed. Two main applications as shown in Figure 6.1, motion segmentation and face clustering are studied in this chapter for this problem.

Errors could exhibit as noise, missed entries, outliers, and sample-specific corruptions in data. Contrary to previous work [97] in this chapter we focus on all types of mentioned errors. To this end, we propose a novel method termed ARPCAC (Approximated Robust Principal Component Analysis Clustering). Given a set of data samples, each of which can be represented as a combination of a low-rank and sparse subspace, ARPCAC aims at finding the *lowest rank* representation of all data jointly, while simultaneously revealing the independent motion of each subspace. The computational procedure of ARPCAC is to solve a *Frobenius* and $\ell_{2,1}$-norm regularised optimisation problem. It can be shown that the ARPCAC can well solve the subspace clustering problem. The subspace membership is provably determined by belonging to either of the low-rank, sparse, or error patterns, and hence the ARPCAC can perform robust subspace clustering and

error correction in an efficient way. Motion segmentation from multiple views has been studied in the case of affine cameras, because in this case the motion of each one of the rigidly moving objects lives in a four-dimensional subspace [38]. In this work however, we do not need to assume an affine camera model, since the camera motion will be compensated for by the dominant subspace that is reasonably close to the background motion in most practical applications.

6.2 Related Work

Existing works can be divided into four main categories: mixture of Gaussian, factorisation, algebraic, and spectral-type methods. Mixture of Gaussian has been used in [64] where a maximum likelihood estimate was used, and in [55] where Random Sample Consensus (RANSAC) was adopted. These methods are sensitive to errors, and this problem is still not well solved due to optimisation difficulty. The main drawbacks of statistical Gaussian methods are that they require the number and dimensions of the subspaces to be known, and they are sensitive to initialisation. Factorisation-based methods [27] seek to approximate the given data matrix as a product of two matrices such that the support pattern for one of the factors reveals the segmentation of the samples. It will be shown that ARPCAC can be regarded as a robust generalisation of the method in [157]. In order to achieve robustness to noise, these methods modify the formulations by adding extra regularisation terms. Nevertheless, such modifications usually lead to non-convex optimisation problems which need heuristic algorithms to solve. Getting stuck at local minima may undermine their performances, especially when the data is grossly corrupted. It will be shown that ARPCAC can be regarded as a robust generalisation of the method in [157]. Generalised Principal Component Analysis (GPCA) [108] presents an algebraic way to model the data drawn from a union of multiple subspaces. This method describes a subspace containing a data point by using the gradient of a polynomial at that point. Then subspace segmentation is made equivalent to fitting the

data with polynomials. GPCA can guarantee the success of the segmentation under certain conditions, and it does not impose any restriction on the subspaces. However, this method is sensitive to noise due to difficulty of estimating the polynomials from real data, which also causes the high computation cost of GPCA. We show that ARPCAC can solve this issue, by robustly discarding the outlier estimates that belong neither to the low-rank subspace nor to the sparse part. Recently, Robust Algebraic Segmentation [128] has been proposed to resolve the robustness issue of GPCA. However, the computational difficulty for fitting polynomials is very large. So these methods can make sense if the data dimension is low and the number of subspaces is small. Our algorithm takes advantage of adopting a low-dimensional representation for the subspaces using ARPCAC and therefore its computation is independent from subspace size. Subspace segmentation has also been regarded as a clustering problem, where an affinity matrix is learned to obtain the final segmentation results by spectral clustering (SC) algorithms such as Sparse Subspace Clustering (SSC) [38], the LRR [97], and the proposed ARPCAC method. The main difference is the approach for learning the affinity matrix. Besides, even if the data is contaminated by outliers, the proposed ARPCAC method is able to recover the low-rank and sparse subspaces, which provably determines the subspace segmentation results. In the presence of arbitrary errors (e.g., corruptions, outliers, and noise), in our experimental evaluations, ARPCAC produces near recovery.

6.3 Low-Rank Modelling of Samples

In this section, we present the ARPCAC method for recovering a matrix from corrupted and incomplete observations. Let D be a collection of data samples in presence of outliers and corruptions. That is, for the set of points $X_p \in P^3$ in frame $f \in F$, we can stack all the image measurements into a $2F \times P$ matrix D as

$$D = \begin{bmatrix} X_1^1 & \dots & X_P^1 \\ Y_1^1 & \dots & Y_P^1 \\ \vdots & \ddots & \vdots \\ X_1^F & \dots & X_P^F \\ Y_1^F & \dots & Y_P^F \end{bmatrix} \tag{6.1}$$

In order to recover the low-rank matrix L from the given observation matrix D corrupted by errors E it is straightforward to consider the following regularised rank minimisation which is similar to (5.4) with the exception that this time the data matrix contains data samples from motion trajectories

$$\min_{rank(L)\leq r,E} \|D - L - E\|_F^2 + \lambda \|E\|_{2,1} \quad s.t. \quad D = L + E, \tag{6.2}$$

where $rank(L) \leq r \ll rank(D)$, and $\lambda > 0$ is a parameter, $\|\cdot\|_F$ is the Frobenius-norm, and $\|\cdot\|_{2,1}$ is the $\ell_{2,1}$-norm which is the ℓ_1-norm of the vector formed by taking the ℓ_2-norm of a matrix. The $\ell_{2,1}$-norm has interesting properties [45], in which it is the maximum value of samples in a group that decides if the group is set to non-zero or not, and it does encourage the rest of the samples to take arbitrary (hence close to maximum) values. The effectiveness of this choice is corroborated with empirical evidence in [45]. This norm definition promotes sparse error patterns more consistent to practical object detection and subspace segmentation than the standard ℓ_1-norm used widely in the literature [17, 97] for this kind of problem, as it helps model the sample-specific corruptions and outliers. The error pattern E can be described as combination of a sparse pattern S containing the underlying subspaces, and a noise pattern G that contains the noise, outliers, and incomplete samples. The main difficulty in subspace clustering problems is the mechanism of dealing with such error patters. ARPCAC is shown to be successful in removing these errors that deviate from the model assumption (subspaces) and the data. The formulation above has been used to achieve the state-of-the-art performance in several applications, however, this formulation implicitly assumes that the underlying data structure is a single low-rank subspace. Moreover, when dealing with samples obtained by a globally moving union of subspaces (e.g., when samples are

taken from a scene with a moving camera and moving objects), this formulation is insufficient. Therefore, we reformulate the problem as

$$\min_{rank(L)\leq r,S,\tau} \|D \circ \tau - L - E\|_F^2 + \lambda\|S\|_{2,1} \quad s.t. \quad D \circ \tau = L + S + G \tag{6.3}$$

where $E = S + G$ and τ stands for some transformation in the image domain (e.g., 2D affine transformation for correcting misalignment, or 2D projective transformation for handling some perspective change in the camera model). And henceforth, $L \circ \tau$ describes the lowest rank estimate for samples drawn solely from the camera motion, whereas S describes all underlying subspaces, and G contains errors. From the sample set $L \circ \tau + S$ we can obtain reliable trajectories for all subspaces. The assumption here is that each subspace has a spectral nature, i.e., each subspace will form a unique affinity matrix that can be used to reveal the true segmentation of data. Also, $L \circ \tau$ provides the underlying lowest rank representation for the data that helps reduce the problem to a simple clustering of independent motions in the scene, as samples S are only drawn from object-induced trajectories. This is the *ideal* case that can happen where the data is clean. There is no loss of generality to assume that the indices of the samples have been rearranged this way to satisfy the true subspace memberships, as the solution produced by ARPCAC is globally optimal and does not depend on the arrangements of the samples. The problem (6.3) is a difficult, non-convex optimisation problem. Fortunately, we can find a good initialisation by pre-aligning all frames in the sequences to the middle frame, before the main loops of minimisation. The pre-alignment is done by the robust multi-resolution method proposed in [119]. This practice is successful in most cases given that a drastic scene change does not occur in the sequence. As described in [158], we can then solve (6.3) by repeatedly linearising about the current estimate of τ, and seeking a deformation step $\Delta\tau$ [124]. In other words, at each iteration, we update τ by a small increment $\Delta\tau$ and linearise $A \circ \tau$ as $D \circ \tau + J\Delta\tau$, where J denotes the Jacobian matrix $J = \frac{\partial D}{\partial \tau}$. Thus, τ can be updated via the following minimisation problem

$$\tau^t \leftarrow \tau + \arg\min_{\Delta\tau} \|D \circ \tau - L^{t-1} - S^{t-1} + J\Delta\tau\|_F^2 \tag{6.4}$$

The minimisation over $\Delta\tau$ in (6.4) is a weighted least-squares problem which has a closed-form solution. In practice, the update of τ for each frame can be done separately since the transformation is applied on each image individually. Thus the update of τ is efficient. Then we proceed by using an alternating minimisation procedure to solve L and S one at a time until the solution reaches convergence and show that it is efficient; that means solving two reduced problems, each being minimised independently form one another

$$L^t = \arg\min_{rank(L)\leq r} \|D \circ \tau - L - S^{t-1}\|_F^2 \tag{6.5}$$

$$S^t = \arg\min_{S} \|D \circ \tau - L^t - S\|_F^2 + \lambda\|S\|_{2,1} \tag{6.6}$$

The residual error of the approximation of D by $L \circ \tau + S$ is stored in G. The entries of G can be very large in magnitude, but random and scattered, exhibiting the behavior of error deviation as described. The discerning difference between S and G is that G shows no structure in the sparsity domain, that of which is determined by the $\ell_{2,1}$-norm minimiser.

6.3.1 Independent subspace motion extraction

The obtained trajectories in S are induced from two motion components: rigid camera motion, and object motion. When the motion of interest includes global object motion, it can be further decomposed into two components: rigid object motion, and articulated motion. We employ the latest advances in sparse optimisation to estimate each of these components, and extract the object trajectories which solely correspond to the motion of interest. [164] and [122] have assumed that the majority of the observed motion is induced by the camera motion; this assumption will not fit most realistic data, so we

refrain from doing so to not cause any loss of generality. Therefore, the trajectories drawn from samples should generally span a subspace determined by the scene structure and the camera's intrinsic and extrinsic parameters. In order to find the basis for the subspace trajectory, we have obtained a $2F \times P$ (P samples) matrix S from ARPCAC using the position vectors of the trajectories in a sequence

$$S = \begin{bmatrix} \mathcal{X}_1^1 & \dots & \mathcal{X}_P^1 \\ \mathcal{Y}_1^1 & \dots & \mathcal{Y}_P^1 \\ \vdots & \ddots & \vdots \\ \mathcal{X}_1^F & \dots & \mathcal{X}_P^F \\ \mathcal{Y}_1^F & \dots & \mathcal{Y}_P^F \end{bmatrix} \tag{6.7}$$

Through the following rank minimisation surrogate, we can decompose S into two components: a low-rank matrix $\mathcal{L}$, and the sparse error matrix $\mathcal{E}$

$$\arg\min_{\mathcal{L},\mathcal{E}} \|\mathcal{L}\|_* + \xi\|\mathcal{E}\|_1 \quad s.t. \quad S = \mathcal{L} + \mathcal{E}, \tag{6.8}$$

with $\|\cdot\|_*$ defining nuclear-norm which is the sum of singular values $\|\mathcal{L}\|_* = \sum_i(\sigma_i)$, and $\|\cdot\|_1$ the ℓ_1 norm. ξ trades off the rank solution versus the sparsity of the error, and is always set to $1.1/\sqrt{P}$ following the theoretical considerations in [17]. The equation (6.8) can be solved with convex optimisation methods such as the Augmented Lagrange Multiplier (ALM) algorithm [95]. The columns of the resulting low-rank matrix $\mathcal{L}$ define the basis of the low-rank components in the trajectories. The subspace spanned by the major basis of $\mathcal{L}$ correspond to the desired background subspace which includes both the background trajectories and the camera motion component of the foreground (objects in the scene) trajectories. On the other hand, any rigid body motions in the scene will also contribute to $\mathcal{L}$; therefore, the subspace spanned by the rest of the basis of $\mathcal{L}$ mostly correspond to rigid body motions. Since the camera motion subspace is approximately spanned by three basis [38, 64], the camera motion component can be estimated by $\mathcal{L}_c = us^*v^T$, where u and v are obtained by singular value decomposition $[u, s, v] = SVD(\mathcal{L})$, and s^* is the top three most significant singular values of s. Therefore, the rigid body motion component is expressed by $\mathcal{L} - \mathcal{L}_c$. Moreover, the columns of the matrix $\mathcal{E}$

correspond to the deviation of each trajectory from the recovered low-rank subspace, which captures the articulated motions [164]. Therefore, the total object trajectories $\mathcal{E}_t$ that include the articulated and the rigid body motion is given by

$$\mathcal{E}_t = \mathcal{E} + \mathcal{L} - \mathcal{L}_c \tag{6.9}$$

If the motion of interest involves only articulated motions without a rigid motion component, (e.g., an object spinning around an axis), the object motion will be mostly captured in $\mathcal{E}$ while the rigid motion component $\mathcal{L} - \mathcal{L}_c$ will be negligible. On the other hand, if the motion of interest involves rigid object motion (e.g., an object moving across the scene), each of $\mathcal{E}$ and $\mathcal{L} - \mathcal{L}_c$ will contribute to the total object motion.

Figure 6.2 shows the motion decomposition for a sequence from the Hopkins155 dataset. As it can be seen, the trajectories that are obtained for the background and foreground, are both contaminated by the camera motion. Note the motion trajectory of the woman walking in the middle column is completely different from the actual motion trajectory that is revealed by ARPCAC in the right column. Clean motion trajectories are crucial for applications such as human motion analysis, and the trajectories in the middle column – which is usually what is obtained by trajectory extractors – would adversely affect the results. Figure 6.3 shows another example with the same scenario, from the Hopkins155 dataset. Here the camera motion influence is more pronounced on the foreground object motion. The woman is walking towards the right of the frame, whereas the extracted motion trajectories do not tell this. Our ARPCAC method is able to both robustly segment the motion clusters while simultaneously compensating for camera motion that is induced upon all subspaces.

In the next example, we show the results for a non-human object moving across the scene, while the camera is moving as well. Figure 6.4 shows such an example where the car in the scene is taking a right-hand corner. As the camera is panning left slowly, this stretches all the foreground object trajectories across the scene. This makes it difficult for

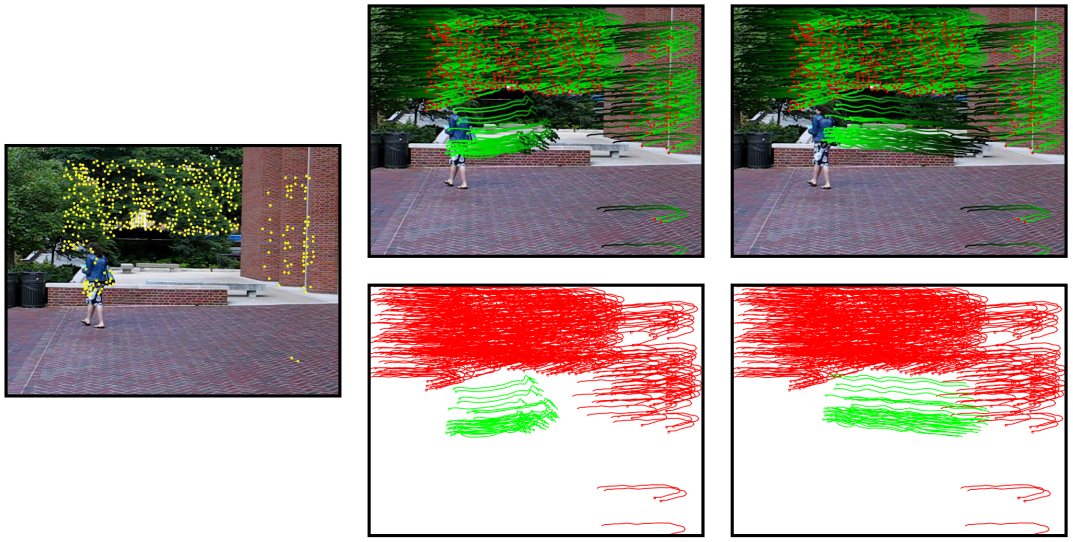


Figure 6.2: An example of independent subspace motion extraction. Left: last frame in a sequence with trajectory particles. Middle: obtained trajectories from tracked motion samples (red corresponds to background subspace trajectories induced by camera motion L, and green corresponds to foreground object trajectories induced by both camera motion and object motion S). Right: extracted clean foreground object trajectories $\mathcal{E}$ in green induced *only* by object motion, revealing the true object trajectory. In the second and third columns motion trajectories of the top figure are shown overlaid over white background in the bottom figure for better visualisation. Please refer to supplementary video at `https://youtu.be/ndE1KZG3yrQ` for more examples.

later processing, to distinguish what the actual object motion was, or even which motion particles correspond to the foreground and which correspond to the background in the scene. ARPCAC is again able to simultaneously cluster the motions, and compensate for the camera panning and zooming out motion.

In the next example we show that ARPCAC can cluster multiple independent motion subspaces. Figure 6.5 illustrates the motion decomposition for three examples in the Hopkins155 dataset. From these examples it is clear that the proposed independent object motion extractor is successful in subtracting camera motion from each motion subspace and simultaneously clustering each motion trajectory into its corresponding subspace.

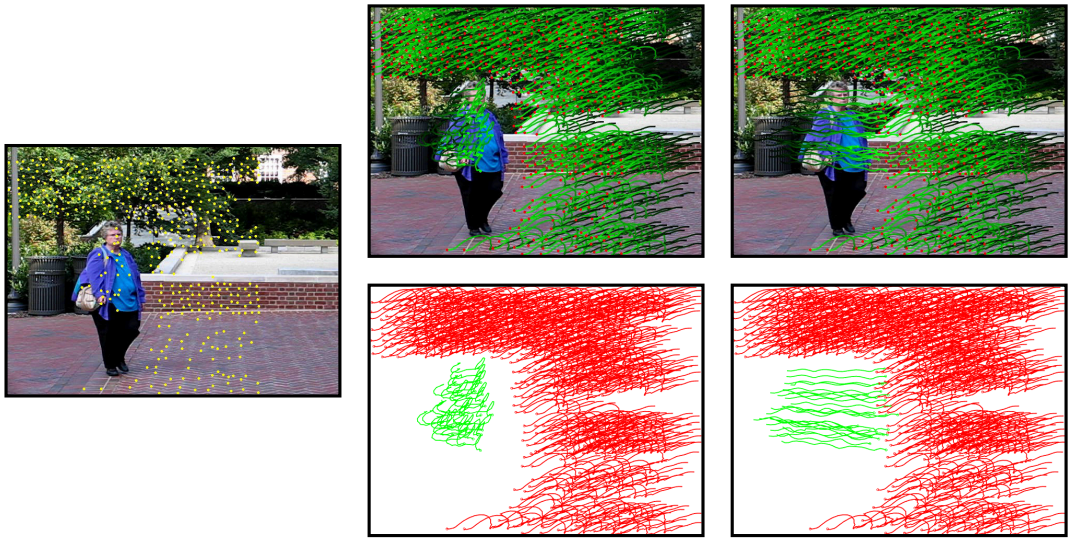


Figure 6.3: An example of independent subspace motion extraction. Left: last frame in a sequence with trajectory particles. Middle: obtained trajectories from tracked motion samples (red corresponds to background subspace trajectories induced by camera motion L, and green corresponds to foreground object trajectories induced by both camera motion and object motion S). Right: extracted clean foreground object trajectories $\mathcal{E}$ in green induced *only* by object motion, revealing the true object trajectory. In the second and third columns motion trajectories of the top figure are shown overlaid over white background in the bottom figure for better visualisation. Please refer to supplementary video at `https://youtu.be/ndE1KZG3yrQ` for more examples.

6.4 Segmentation of Multiple Rigid-Body Motions

From the geometry of the 3D motion segmentation problem from multiple affine views, one can assume that the problem of multi-frame motion segmentation is equivalent to clustering multiple low-dimensional linear subspaces of a high-dimensional space. The Costeira and Kanade's multibody factorisation algorithm [27] fails when the motion subspaces are not independent. From here on we regard the problem of multi-frame motion segmentation with an approach that works for all the spectrum of affine motions: from two-dimensional and partially dependent to four-dimensional and fully independent. This is achieved by a combination of our ARPCAC method and PowerFactorisation that leads to the following geometric solution to the multi-frame 3D motion segmentation

Figure 6.4: An example of independent subspace motion extraction. Left: last frame in a sequence with trajectory particles. Middle: obtained trajectories from tracked motion samples (yellow corresponds to background subspace trajectories induced by camera motion L, red and green correspond to foreground object trajectories induced by both camera motion and object motion S). Right: extracted clean foreground object trajectories $\mathcal{E}$ in red and green induced *only* by object motion, revealing the true object trajectory. In the second and third columns motion trajectories of the top figure are shown overlaid over white background in the bottom figure for better visualisation. Please refer to supplementary video at `https://youtu.be/ndE1KZG3yrQ` for more examples.

problem [157]:

1. Project the motion trajectories obtained from object-induced motion $\mathcal{E}$ extracted by ARPCAC onto a five-dimensional subspace using the PowerFactorisation.

2. Fit a collection of subspaces to the projected trajectories:

   (a) Fit a homogeneous polynomial representing all motion subspaces to the projected data.

   (b) Obtain a basis for each motion subspace form the derivatives of this polynomial.

(c) Apply spectral clustering to a similarity built form the subspace angles to cluster the data.

The above assumptions are reasonable as in contrast to [157] we do not deal with *complete data*, since this is unrealistic in most real data; in addition we do not need to handle *data with outliers*, as the outliers are stored in matrix G that is the result of the decomposition in ARPCAC. Assuming that the data is maximally 5-dimensional is also sound, since we have discarded all non-rigid camera motions in the matrix $L \circ \tau$. If we first project the object induced-motions $\mathcal{E}$ onto $\mathbb{R}^5$, the motion subspaces become partially dependent, because the rank of the projected data matrix is at most 5 [157]. The reason for projecting is that the segmentation of data lying in multiple subspaces is preserved by a generic linear projection. For instance, if one is given data lying in two lines in $\mathbb{R}^3$ through the origin, then one can project the lines onto a plane in general position and then cluster the data inside that plane. The same principle applies to the motion segmentation problem. Since we know that the maximum dimension of each motion subspace is four, then projecting onto a generic five-dimensional subspace preserves the clustering of the motion subspaces. In order for two motion subspaces to be distinguishable form each other, it is enough for them to be different along one dimension, i.e., we do not really need them to be different in all four dimensions. It is the key observation the one that enables us to treat *all* partially dependent motions as well as *all* independent motions in the same framework: clustering subspaces of dimension two, three, or four living in $\mathbb{R}^5$. Another advantage of projecting the data onto a 5-dimensional space is that, except for the projection itself, the complexity of the motion segmentation algorithm we are about to present becomes independent of the number of frames. Indeed, our algorithm would require a minimum of only *three* frames for *any* number of independent motions. We have tested our approach on a database of 155 motion sequences with full, independent, degenerate, dependent motions, missing data, outliers, etc. Our algorithm achieves error of 0.89% for two motions and 3.78% for three motions.

6.4.1 Projection using PowerFactorisation

From here on, without loss of generality, we refer to the sample matrix as W that could refer to either object samples S or object-induced samples $\mathcal{E}$. We can use the technique of PowerFactorisation [70], [155], which in some cases may be more rapid than the contender SVD in [157]. In addition, it allows to deal with the case in which some entries of the data matrix $W \in \mathbb{R}^{2F \times P}$ are missing[1]. Clearly this cannot be done with SVD. We use a method adapted for incomplete data, based on an analysis of the "power method" for computation of eigen-values of a matrix. PowerFactorisation gives a rapid method for approximating low-rank matrices and is discussed in detail in [70], [155]. We wish to replace W by a matrix obtained by projecting its columns onto a 5-dimensional subspace. If AB^T is the nearest rank-5 factorisation to W, then $\hat{W} = B^T$ is the matrix that we require. The measure of closeness of AB^T to W is

$$\sum_{(i,j)\in\mathcal{I}} (W_{ij} - (AB^T)_{ij})^2, \tag{6.10}$$

where $\mathcal{I}$ is the set of pairs (i, j) for which W_{ij} is known. With PowerFactorisation we start with a random matrix A_0, and alternate the following steps until convergence of $A_k B_k^T$. Essentially this algorithm alternates between computing A_k and B_k using least-squares.

1. Given A_{k-1}, find the $P \times r$ matrix B_k that minimises $\sum_{(i,j)\in\mathcal{I}} |W_{ij} - (A_{k-1}B_k^T)_{ij}|^2$.

2. Orthonormalise the columns of B_k by replacing it by a matrix B'_k such that $B_k = B'_k N_k$, where B'_k has orthonormal columns, and N_k is upper-triangular.

3. Given B_k, find the matrix A_k that minimises $\sum_{(i,j)\in\mathcal{I}} |W_{ij} - (A_k B_k^T)_{ij}|^2$

The computation of each B_k and A_k proceeds just one column at a time, and consists of finding the least-squares solution to a set of linear equations.

[1] A fairly common occurrence in feature tracking due to occlusions or points disappearing from the field of view.

6.4.2 Fitting polynomials to projected trajectories

We have reduced the motion segmentation problem to finding a set of linear subspaces in $\mathbb{R}^5$, each of dimension at most 4, which contain the data points (or come close to them). The points in question, $\{w_p\}_{p=1}^{P}$, are the columns of the projected data matrix $\hat{W} = [w_1, \ldots, w_P] \in \mathbb{R}^{5\times P}$. We obtain a polynomial q representing the n motion subspaces by computing its vector of coefficients $c \in \mathbb{R}^{M_n}$ as the singular vector of the embedded data matrix $\tilde{W} = [\tilde{w}_1, \ldots, \tilde{w}_P] \in \mathbb{R}^{M_n \times P}$ corresponding to its smallest singular value.

6.4.3 Feature clustering via polynomial differentiation

The feature points can then be clustered by applying spectral clustering to the similarity matrix $\mathcal{S}_{ij} = \cos^2(\theta_{ij})$, where θ_{ij} is the angle between the vectors $\nabla q(w_i)$ and $\nabla q(w_j)$ for $i, j = 1, \ldots, P$, with the derivative of q defined as a 5-vector

$$\nabla q(w) = (\partial q/\partial w_1, \ldots, \partial q/\partial w_5) \tag{6.11}$$

Then the standard factorisation approach is applied to each one of the n group of features to obtain motion and structure parameters.

6.5 Experiments

In the experiments of this chapter, we focus on analyzing the essential aspects of ARP-CAC under the context of subspace segmentation and outlier detection. We have implemented our algorithm in MATLAB R2015a on a desktop machine with a Core i7-4770 (single core). For all the tests we chose $\lambda = 5 \times 10^{-3}$. We compare our method to some previous subspace segmentation methods, including Random Sample Consensus (RANSAC) [55], Generalised PCA (GPCA) [108], Local Subspace Analysis (LSA) [167], Agglomerative Lossy Compression (ALC) [127], Sparse Subspace Clustering (SSC) [38],

Spectral Curvature Clustering (SCC) [21], Multi Stage Leaning (MSL) [144], Locally Linear Manifold Clustering (LLMC) [61], Local Best-fit Flats (LBF) [173], Low-Rank Representation (LRR) [97], Low-Rank Representation Heuristic (LRR-H) [97], LRSC [52], RPCA methods from RPCA_1 [17], $\text{RPCA}_{2,1}$ [166], and [138], SR [33], Spectral LBF (SLBF) [173], BDLRR [54], and the most recent work S^3C [89].

6.5.1 Hopkins155

To verify the segmentation performance of ARPCAC, we adopt for experiments the Hopkins155 [153] motion database, which provides an extensive benchmark for testing various subspace segmentation algorithms. In Hopkins155, there are 155 video sequences along with the features extracted and tracked in all the frames. Each sequence is a sole dataset (i.e., data matrix) and so there are in total 155 datasets of different properties, including the number of subspaces, the data dimension, and the number of data samples. Although the outliers in the data have been manually removed, some sequences are grossly corrupted and have notable error levels. The segmentation performance for this dataset is shown in Table 6-A. These results illustrate that ARPCAC performs considerably better than other PCA-based counterparts, namely PCA, RPCA_1, $\text{RPCA}_{2,1}$, SR, LRR, and GPCA. Besides the superiority in segmentation accuracy, another advantage of ARPCAC is that it can work well under a wide range of parameter settings as we chose the same λ value for all the test, whereas other PCA-based methods except for LRR are sensitive to the parameter λ. As for comparison to state-of-the-art methods in the lower tier of Table 6-A, our method performs on par, and achieves third place after SSC and SLBF. This performance can be improved if λ is tuned per-problem, but we refrain from doing so as we would like to demonstrate an autonomous performance. Moreover, our algorithm is superior in clustering multiple motions in a scene as shown in Table 6-C, whereas, SSC and SLBF both are more well-suited for single motion segmentation. The efficiency in terms of running time of ARPCAC is comparable to PCA and surpasses that of other PCA-based methods as shown in Table 6-B, making it suitable for

Table 6-A: Segmentation Errors (%) on Hopkins155.

	PCA	$RPCA_1$	$RPCA_{2,1}$	SR	LRR	GPCA	RANSAC	BDLRR	**ARPCAC**
mean %	4.56	4.13	3.26	3.89	1.59	10.34	9.76	4.33	1.53
	LLMC	LBF	ALC	SCC	SLBF	SSC	MSL	S^3C	LSA
mean %	4.80	3.72	3.37	2.70	1.35	**1.24**	5.06	2.20	4.94

Table 6-B: Average run time (seconds) per sequence for segmentation task on Hopkins155 for RPCA-based methods.

PCA	$RPCA_1$	$RPCA_{2,1}$	SR	LRR	**ARPCAC**
0.2	0.8	0.8	4.2	1.9	0.4

Table 6-C: Clustering Error (%) of Different Algorithms on Hopkins155 for 2 and 3 motions.

	LSA	SCC	LRR	LRR-H	LRSC	SSC	S^3C	BDLRR	**ARPCAC**
2 Motions									
mean %	3.61	3.04	4.83	3.41	3.87	1.83	1.64	3.70	**0.89**
median %	0.51	**0.00**	0.26	**0.00**	0.26	**0.00**	**0.00**	**0.00**	**0.00**
3 Motions									
mean %	7.65	7.91	9.89	4.86	7.72	4.40	4.11	6.49	**3.78**
median %	1.27	1.14	6.22	1.47	3.80	**0.00**	0.73	1.20	1.31
All									
mean %	4.52	4.14	5.98	3.74	4.74	2.41	2.20	4.33	**1.53**
median %	0.57	**0.00**	0.59	**0.00**	0.58	**0.00**	**0.00**	**0.00**	**0.00**

real-time performance. Theoretically, the computational complexity of ARPCAC is the same as RPCA methods. ARPCAC costs more computational time than PCA because its transformation parameter estimation for the dominant subspace needs iterations to converge. The results of applying subspace clustering algorithms to the dataset using the original $2F$-dimensional feature trajectories for 2-motion and 3-motion categories on Hopkins155 are shown in Table 6-C. Our algorithm achieves top performance in all motion categories.

6.5.2 Yale-Caltech

To test ARPCAC's effectiveness in the presence of outliers and corruptions, we create a dataset by combining the Extended Yale Database B [88] and Caltech101 [53]. For comparison to prior works, for Extended Yale Database B, we remove the images pictured

Table 6-D: Segmentation Accuracy (ACC) and time consumption comparison on Yale-Caltech for PCA-based methods.

	PCA	RPCA_1	$\text{RPCA}_{2,1}$	SR	LRR	**ARPCAC**
Accuracy %	77.15	82.97	83.72	73.17	86.13	**89.22**
time (seconds)	**0.6**	60.8	59.2	383.5	152.6	53.87

under extreme light conditions. Namely, we only use the images with view directions smaller than 45° and light source directions smaller than 60°, resulting in 1204 authentic samples approximately drawn from a union of 38 low-rank subspaces (each face class corresponds to a subspace). For Caltech101, we only select the classes containing no more than 40 images, resulting in 609 non-face outliers. Figure 6.6-Left shows some examples of this dataset. It can be seen in Table 6-D that ARPCAC is better than PCA and RPCA methods in terms of both subspaces segmentation and outlier detection. To visualise ARPCAC's effectiveness in error correction, Figure 6.6-Right shows some produced results. It is worth noting that the "error" term E can contain "useful" information, e.g., eyes and salient parts, that can be used for emotion and visual cue recognition. The low-rank part $L \circ \tau$ corresponds to the principal features of each subject that discriminate it from the rest of the data. The aligned and cleaned $L \circ \tau$ part can be used for face recognition, and face clustering as done in this chapter.

6.5.3 LFW

For a more challenging and uncontrolled test on effectiveness of ARPCAC in presence of severe misalignment, outliers, and corruptions we use realistic examples taken from the Labelled Faces in the Wild (LFW) database of public figures [76]. These images exhibit significant variations in pose and facial expression, illumination, and occlusion; moreover, the ground truth (i.e., undistorted, not rotated, not shifted) image is not known. In total there are 681 samples of images taken from 20 subjects. Our ARPCAC aligns these images to a 80×60 canonical frame, and Affine transformations τ are used to cope with large variability in poses. Figure 6.7 shows one example from the results on this dataset. Our algorithm proves itself to be effective even in presence of large

misalignment and corruptions.

6.6 Conclusion

We have proposed a low-rank and sparse representation to identify the subspace structures from corrupted data. Namely, our goal is to segment the samples into their respective subspaces and correct the possible errors simultaneously while revealing each subspace's independent motion. ARPCAC is a generalisation of the recently established RPCA method [17], extending the recovery of corrupted data from single subspace to multiple subspaces that are dynamic where both camera and the scene objects move. Both theoretical and experimental results show the effectiveness of ARPCAC in subspace segmentation and misaligned and corrupted face clustering applications.

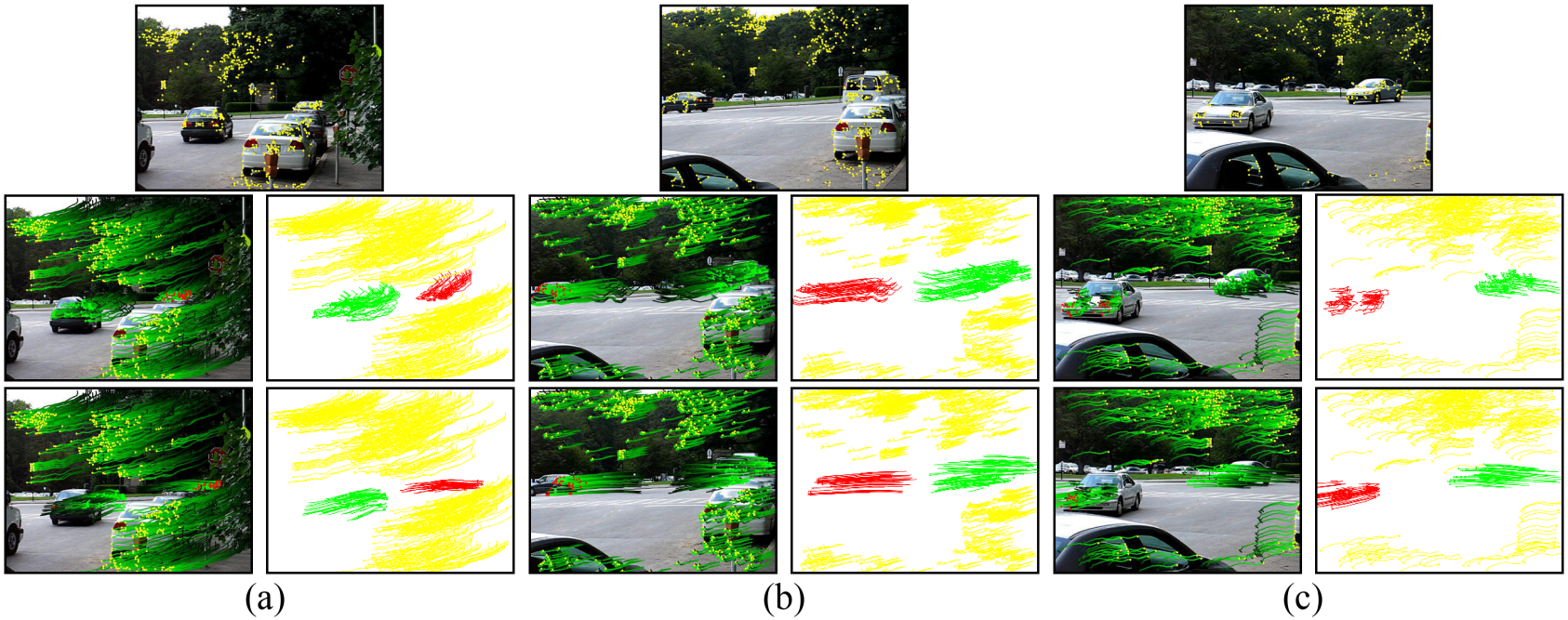


Figure 6.5: Three examples of independent subspace motion extraction (a), (b), and (c). Top: last frame in a sequence with trajectory particles. Middle: obtained trajectories from tracked motion samples (yellow corresponds to background subspace trajectories induced by camera motion L, red and green correspond to foreground object trajectories induced by both camera motion and object motion S). Bottom: extracted clean foreground object trajectories $\mathcal{E}$ in red and green induced *only* by object motion, revealing the true object trajectory. For each example, motion trajectories of the left figure are shown overlaid over white background in the right figure for better visualisation. Please refer to supplementary video at `https://youtu.be/ndE1KZG3yrQ` for more examples.

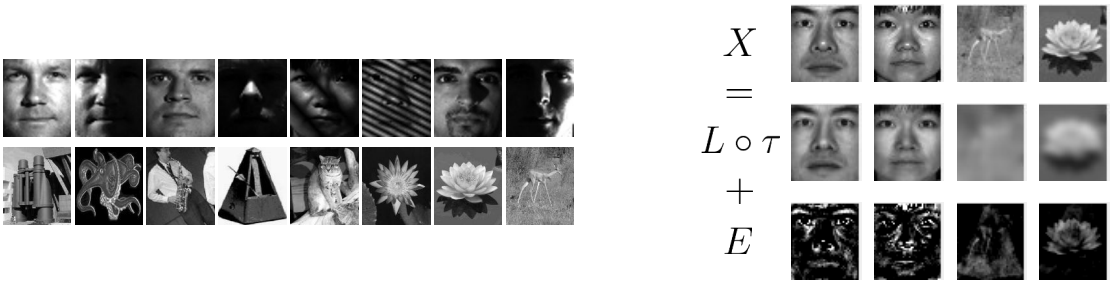


Figure 6.6: Left: examples of the images in the Yale-Caltech dataset as used in [97]. Right: some examples of using ARPCAC to correct the errors in the Yale-Caltech dataset; from top to bottom: the original data matrix X, the corrected data $L \circ \tau$, the error E.

Figure 6.7: An example from the LFW database. Left: original images D; middle: aligned images $D \circ \tau$; right: errors E.

Chapter 7

Video Super-Resolution

Sparse coding-based algorithms have been successfully applied to the single-image super resolution (SR) problem. Recently, these algorithms have been extended to the multiple-image case improving the reconstruction quality. When processing video information it is reasonable to assume that most of the content in a frame is shared by neighbouring frames. Conventional multi-image SR algorithms incorporate auxiliary frames into the model by a registration process using subpixel block matching algorithms that are computationally expensive. There is a need for a mechanism to incorporate the spatio-temporal information in an SR algorithm. This becomes increasingly important as super-resolving UHD video content with existing sparse-based SR approaches becomes less efficient. In order to fully utilise the spatio-temporal information, where one frame of the video is super-resolved from multiple neighbouring frames, we propose a novel multi-frame video SR approach that is aided by a low-rank plus sparse decomposition of the video sequence. We introduce a group of pictures (GOP) structure where we seek a rank-1 low-rank part that recovers the shared spatio-temporal information among the frames in the GOP. Then we super-resolve the low-rank frame and sparse frames separately, and use the high-resolution versions of these to reconstruct the SR video. This assumption results in significant time reductions for calculating a SR video in the sparse coding frame-

work. Extensive experimental evaluation demonstrates the effectiveness of our approach in outperforming current state-of-the-art SR methods both qualitatively and in terms of complexity. The findings of this chapter are published in [43], [44].

7.1 Introduction

We denote the LR image as Y, and the HR image of the same scene as X. Lowercase y and x denote the low- and high- resolution image patches, respectively. D is used to refer to the dictionary for sparse coding; specifically the D_l and D_h denote the dictionaries for low- and high- resolution image patches, respectively. It has been statistically proven that image patches can be well-represented as a sparse linear combinations of elements, namely atoms of a dictionary taken from a finite and not too big bag [32], [168]. Each vectorised patch $y \in \mathbb{R}^m$ of an LR image Y, can be written as:

$$y = \alpha_1 D_1 + \alpha_2 D_2 + \cdots + \alpha_n D_n, \tag{7.1}$$

where most of the coefficients $\alpha_1, \alpha_2, \ldots, \alpha_n$ are zero if the atoms $D_1, D_2, \ldots, D_n$ of the dictionary D are properly selected. When $m = n$, D has to be a complete basis to represent any patch. However, when $n > m$ it is possible to find solutions $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ where a considerable number of coefficients α_i are zero. We can conveniently assume a sparse representation for y as each patch is completely determined for a substantially reduced number of parameters that is usually far less than the number of atoms.

To calculate the sparse representation of a patch one needs to determine the appropriate dictionaries D (learning phase), and then estimate the coefficients of the linear combination of the atoms (testing phase). We can find the sparsest α results in the convex Lasso regularised minimisation problem below

$$\min_{\alpha} \|D\alpha - y\|_2^2 + \mu\|\alpha\|_1, \tag{7.2}$$

where μ is a regularisation parameter to balance the reconstruction error and sparsity. Different solvers such as Least Angle Regression (LARS), Shooting Algorithm, etc., have been used to solve this problem. A systematic way to calculate the dictionary D is solving the following minimisation problem

$$\min_{D,Z} \|DZ - X\|_2^2 + \mu\|Z\|_1, \tag{7.3}$$

where X is the HR training data. The objective function above is non-convex with respect to both D and Z. Z contains the coefficients of the linear combination of the atoms that approximate the training data. The problem above can be solved in an alternating process, by keeping one fixed and solving for the other at a time until convergence. This alternating solution is convex. The selection of training data and the incorporation of structures and characteristics in the dictionaries is application-specific.

7.2 Single-Image SR based on Sparse Coding

Yang *et al.* [168] assume that the degradation from the HR patch x to the LR patch y is nearly linear, where each HR patch and its corresponding LR patch share the same sparse linear coefficients $\alpha = (\alpha_1, \ldots, \alpha_n)$. The high-resolution dictionary D_h and the low-resolution dictionary D_l need to be defined properly. There are then two stages to solving the sparse representation-based SR: the learning phase where the bi-level dictionaries D_h and D_l are constructed, and the testing phase where the vector coefficients α that correspond to each LR patch are calculated.

7.2.1 Learning phase

We assume that the sparse representation of the HR patches is the same as the sparse representation of the corresponding LR patch; therefore, the set of training samples can be formed by a group of N HR sampled patches X_h and M LR sampled patches Y_l. The

HR and LR vectorised atoms are the columns of the matrices D_h and D_l that solve the following minimisation problem

$$\min_{D_h, D_l, Z} \|X_c - D_c Z\|_2^2 + \mu \|Z\|_1, \tag{7.4}$$

where $X_c = \begin{bmatrix} \frac{1}{\sqrt{N}} X_h \\ \frac{1}{\sqrt{M}} Y_l \end{bmatrix}$ and $D_c = \begin{bmatrix} \frac{1}{\sqrt{N}} D_h \\ \frac{1}{\sqrt{M}} D_l \end{bmatrix}$. Here $N = M$.

The minimisation problem above is non-convex with three variables D_h, D_l and Z. A convex solution would be an alternating process where two variables are kept fixed and the other one is solved until convergence. When D_h and D_l are fixed, the optimisation problem is solved by non-negative quadratic linear programming using feature sign (L1QP solver). When Z is fixed, a constrained quadratic programming technique in its dual formulation is used. The details of this solution appear in [87].

7.2.2 Testing phase

Here, given a LR patch y, the HR desired patch x can be defined as

$$x = D_h \alpha^l, \tag{7.5}$$

where α^l is the solution of the minimisation problem

$$\alpha^l = \arg\min_{\alpha} \|y - D_l \alpha\|_2^2 + \mu \|\alpha\|_1 \tag{7.6}$$

This problem can be solved using the LARS-Lasso algorithm [35] or the feature-sign search algorithms [87]. To increase perceptual quality of the results a few more steps are required. In order to enforce the compatibility between adjacent patches, the authors in [168] proposed an overlapping strategy that modifies the minimisation problem (7.6) that involves the HR and LR dictionaries. Also, a feature transformation F is used to

enforce the high-frequency content of the LR image. Finally, once the HR image has been reconstructed patch by patch using sparse coding, a back-projection algorithm is performed to enforce the global reconstruction constraint to correct for noise in the LR image.

7.3 VSRGOP: Multi-Frame Video SR

We propose a novel sparse coding-based algorithm for multi-frame SR in videos that is aided by a low-rank and sparse decomposition (LRSD) to fully utilise the spatio-temporal information in the video. To the best of our knowledge only a handful of algorithms based on multi-frame sparse coding-based SR exist in the literature where usually an expensive block-matching algorithm is used. Our algorithm is the first to involve a LRSD step in order to avoid the registration by block-matching. The majority of SR algorithms have been proposed to the SISR problem and do not take into account the temporal information in videos. In [175] the authors proposed to use the motion vectors, block sizes, and prediction residual that is computed by the video encoder in compressed videos to accelerate their algorithm. Low-rank and sparse decomposition (LRSD) methods have been used in many applications such as background subtraction [45], [40], robust subspace clustering, etc.; however, these LRSD models are not suitable for the problem at hand. To adapt the LRSD to the SR problem, we propose a novel modified approximated RPCA model, and an efficient alternative SVD-free approximated RPCA where the low-rank component L is a rank-1 matrix and the sparse matrix S has a tree-regularised block structure.

As discussed before, the main limitation of using the sparse coding-based algorithms for video SR is the high computational cost associated with the super-resolving frames individually. Here, we propose a novel approach that alleviates the high computational cost. Our method obtains greater visual quality while achieving significant reduction of the number of floating point operations.

We propose to super-resolve the LR video in GOPs of F frames with $F = [8, 16, 24, 32, 64]$; we decompose each GOP into a low-rank component L that contains mostly the static unchanging parts of the scene and a sparse component S that contains dynamic pixels, changes in the scene, and possible noise. Then each obtained L and S image for the frames in the GOP are upsampled separately using the sparse coding method described in the previous section. Notice that since we perform the SR on a low-rank component that is obtained by decomposing a GOP, we implicitly incorporate temporal information into our SR approach. Another advantage of this method is that, since the sparse component S is expected to contain very few non-zero blocks of pixels, the upsampling for each sparse image can be performed with several orders of magnitude faster than that of a non-sparse image. Therefore, the spatio-temporal information in the GOP are fully exploited without having to calculate any block matching, complex registration, or relying on motion vectors calculated by the video encoder. Then the shared information between the images in the GOP that is contained in the matrix L is upsampled only once – again providing time savings – as opposed to having to perform the upsampling for each frame individually. The LRSD provides a robust motion compensation possibility for the cases where camera-induced motion is present in the video sequence. The assumption of low-rankness and sparsity itself gives a good cue for being able to describe the global motion in the scene as transformations between the low-rank images in adjacent frames. We find that in videos containing camera-induced motion, our method performs better than the state-of-the-art alternatives.

7.3.1 LRSD for SR problem

Given a set of frames in a GOP of N frames $I = \{I_1, I_2, \ldots, I_n\}$, we can form the matrix $A \in \mathbb{R}^{m \times n}$ by stacking the frames in I as columns in the matrix A. The problem of finding a low-rank matrix L and a sparse matrix S such that $A = L + S$ has been extensively studied in the literature [17], [176], [124], [40], [45]. In [45], the authors propose a modified approximated RPCA where they solve a 3-term decomposition problem. Similar

to (5.11), we are interested in decomposing the matrix A into 2 terms L and S as

$$\min_{rank(L)\leq r,S,\tau} \|A \circ \tau - L - S\|_F^2 + \lambda\psi(S) \tag{7.7}$$

where we have strictly set $rank(L) \leq r \leq rank(A)$. $\|\cdot\|_F$ is the Frobenius norm of a matrix defined as $\|A\|_F = \sqrt{\sum_{i,j} A_{ij}^2}$; λ is a scalar that controls the amount of data in S. We find that setting it to $\lambda = 1/\sqrt{max(m,n)}$ works well for our experimental data. τ stands for some transformation describing the global motion induced by camera motion (e.g. 2D affine transformations, or 3D projective transformations).

The matrix S contains noise and sparse components. Similar to Chapter 5 we use a tree-structured sparse component since it better describes the spatial connectivity of the pixels in the sparse matrix. We explained in section 5.5 that the scene in a frame can be described using a tree structure by subdivision where each child node is a subset of its parent node and the nodes of the same depth level do not overlap. Hence, similar to before denote $\mathcal{G}$ as a set of groups from the power set of the index set $\{1,\ldots,m\}$, with each group $G \in \mathcal{G}$ containing a subset of these indices. The aforementioned tree-structured groups used in this chapter are formally defined as follows: A set of groups $\mathcal{G}$ is said to be *tree-structured* in $\{1,\ldots,m\}$ if $\mathcal{G} = \{\ldots, G_1^i, G_2^i, \ldots, G_{b_i}^i, \ldots\}$ where $i = 0, 1, 2, \ldots, d$, d is the depth of the tree, $b_0 = 1$ and $G_1^0 = \{1, 2, \ldots, m\}$, $b_d = m$ and correspondingly $\{G_j^d\}_{j=1}^m$ are singleton groups. Let G_j^i be the parent node of a node $G_{j'}^{i+1}$ in the tree, we have $G_{j'}^{i+1} \subseteq G_j^i$. We also have $G_j^i \cap G_k^i = \emptyset, \forall i = 1,\ldots,d, j \neq k, 1 \leq j,k \leq b_i$. Similar group structures are also considered in [45], [81]. With the above notation, a general tree-structured sparsity-inducing norm can be written as

$$\psi(S) = \sum_{i=0}^{d}\sum_{j=1}^{b_i} w_j^i \|S_{G_j^i}\|_{2,1}, \tag{7.8}$$

where $S_{G_j^i}$ is a vector with entries equal to those of S for the indices in G_j^i and 0 otherwise. w_j^i are positive weights for groups G_j^i chosen as $w_j^i = 1/max(A_{G_j^i})$ to enforce illumination invariance in the regularisation scheme across patches. The regulariser $\psi(\cdot)$ on S is chosen to be $\|\cdot\|_{2,1}$. $\ell_{2,1}$-norm is a group sparsity inducing norm that acts in a tree-structured which involves a hierarchical partition of the m variables in S into

groups.

The optimisation problem (7.7) is solved via an alternating minimisation strategy described in [45]. First an initialisation of τ is found, by pre-aligning all the frames in the GOP to the middle frame. Then τ is linearised via the robust multiresolution method proposed in [119], [124]. Then the function is minimised for L and S separately until convergence in a similar fashion as (5.14) and (5.15) with the following subproblems

$$L^t = \underset{rank(L)\leq r}{\arg\min} \|A \circ \tau - L - S^{t-1}\|_F^2 \tag{7.9}$$

$$S^t = \arg\min_S \|A \circ \tau - L^t - S\|_F^2 + \lambda \sum_{i=0}^{d} \sum_{j=1}^{b_i} w_j^i \|S_{G_j^i}\|_{2,1} \tag{7.10}$$

Both these subproblems have non-convex constraints. Their global solutions L^t and S^t exist. In particular, the two subproblems can be solved by updating L^t via singular value hard thresholding of $A - S^{t-1}$ [176], and updating S^t via our structured-sparsity inducing norms with a soft-thresholding with λ. The penalty term in (7.10) assures the structured-sparsity of S w.r.t. the defined tree-structured groups.

7.3.2 Modified SVD-free LRSD

In the LRSD we can conveniently assume that the background for a GOP can be described only by one frame, as we do not expect very drastic changes within the defined GOP sizes in our framework. We strictly set $rank(L) = 1$, thus the expensive SVD calculation for background estimation seems to be unnecessary. Here, we present an alternative approach to recovery of a rank-1 matrix from the data matrix A. Assume the matrix L as $L = l\mathbb{1}^T$ where l is a vector and $\mathbb{1}^T = [1, 1, \ldots, 1]$ and the same length as l, i.e., L would be a matrix with identical columns. Then, the minimisation problem

to be solved is

$$\arg\min_{l,S,\tau} \|A \circ \tau - l \times \mathbb{1} - S\|_F^2 + \lambda \sum_{i=0}^{d} \sum_{j=1}^{b_i} w_j^i \|S_{G_j^i}\|_{2,1} \tag{7.11}$$

Following the same alternative minimisation strategy as before, the subproblem to be solved in each iteration is now modified to

$$l^t = \arg\min_{l} \|A \circ \tau - l \times \mathbb{1} - S^{t-1}\|_F^2 \tag{7.12}$$

Denoting $E = A \circ \tau - S^{t-1}$, the following Lemma gives a closed-form solution for the vector l where the SVD calculation of A is not needed.

Lemma 1. *The solution l of the optimisation problem* $\arg\min_l \|E - l\mathbb{1}^T\|_F^2$ *is given by:*

$$l_i = \frac{1}{n} \sum_{j=1}^{n} E_{ij} \quad , \quad i = 1, \ldots, m$$

Proof. Expanding the objective function we have:

$$\begin{aligned}
\|E - l\mathbb{1}^T\|_F^2 &= \sum_{k=1}^{n} \|E_k - l\mathbb{1}^T\|_F^2 \\
&= (E_{11} - l_1)^2 + \cdots + (E_{m1} - l_m)^2 + \ldots \\
&+ (E_{1n} - l_2)^2 + \cdots + (E_{mn} - l_m)^2 \\
&= (E_{11} - l_1)^2 + \cdots + (E_{1n} - l_1)^2 + \ldots \\
&+ (E_{m1} - l_m)^2 + \cdots + (E_{mn} - l_m)^2 \\
&= \sum_{i=1}^{m} \sum_{j=1}^{n} (E_{ij} - l_i)^2
\end{aligned}$$

Algorithm 5 VSRGOP Algorithm

Input: LR frames of the GOP
Output: HR frames of the GOP
Learning phase: Construct the bilateral dictionaries D_h and D_l following the strategy by [168]. (This phase can be performed in advance and use D_h and D_l as inputs of the algorithm.)
Testing phase:

1) Estimate the LRSD of matrix A, while estimating the camera motion as $A \circ \tau \approx L + S$, where $rank(L) = 1$, S is block-sparse, and τ is the transformation parameter.
2) Construct a HR version of the frame corresponding to background frame using the SISR algorithm described in Section 7.2.
3) For all the frames in the GOP $(1, 2, \ldots, N)$ construct a HR version of the frames corresponding to the columns S using the SISR algorithm.
4) Reconstruct the SR version of the GOP with the HR background and HR foreground frames, applying the inverse transformation.

Setting the derivatives of each i-th term $\sum_{j=1}^{m} (E_{ij} - l_i)^2$ to zero with respect to l_i yields:

$$\begin{aligned} \sum_{j=1}^{n} (E_{ij} - l_i)^2 &= 0 \\ -2(E_{i1} - l_i) - 2(E_{i2} - l_i) - \cdots - 2(E_{in} - l_i) &= 0 \\ -2(E_{i1} + \cdots + E_{in}) + 2nl_i &= 0 \end{aligned}$$

We have from there that:

$$l_i = \frac{1}{n} \sum_{j=1}^{n} E_{ij}$$

■

Using the LRSD method we propose the VSRGOP algorithm, shorthand for *Video Super Resolution using Groups of Pictures*. The parameters that we need to set for this algorithm are: number of atoms of the dictionaries, patch size, number of frames in GOP, the overlap size of patches, regularisation parameter, and scale factor. Algorithm 5 describes VSRGOP steps in detail. Following the strategy in [168], in steps 2 and 3 of Algorithm 5 we use a high-pass filtering in order to extract local features that correspond to the high-frequency content. Also, a back-projection step is performed as part of both these steps. Where the back-projection is used in our tests we refer to it as VSRGOP + BP. In step 4 the HR backgorund and HR foreground frames are simply added to create the SR video.

7.4 Experiments

In this section we show a comparative study of the performance of the proposed algorithm for video and single image SR. We first demonstrate the SR results obtained by applying our method on video sequences from our test databases. Then we show that our method can be successfully applied to the SISR problem despite being a video SR algorithm by nature. Finally we move on to discuss how various influential factors for the proposed algorithm affect the global reconstruction, as well as the computational complexity. For video super-resolution we use the following datasets: **BBC**[1], **Ultra Video Group (UVG)**[2], and **SJTU**[3] [139]. These three datasets comprise of 27 videos of 10 seconds each at 60fps. For our tests we use all the frames in the videos. Since by default we choose GOP size of 8 frames, we report average results for an 8-frame GOP where applicable. For single image super-resolution we use the publicly available **Set5**[4] and **Set14**[5] datasets. Our algorithm is implemented in MATLAB and run on a Core i7-4770 CPU @3.40GHz (single core) and 32GB of RAM. We compare our method against state-of-the-art in sparse coding SR methods, namely Kato *et al.* [84], Yang *et al.* [168], and a state-of-the-art deep learning approach by Dong *et al.* [31], as well as the baseline Bicubic interpolation. We set the parameters of our algorithm for these experiments as:

- *Dictionaries*: The dictionaries D_h and D_l are learned using 100,000 patches extracted from 57 HR natural images. The number of atoms is 512 or 1024. Following other papers, we use filters to extract the features from the upsampled version of the LR images. In our tests, we set the number of atoms in the dictionary to 512 as default, unless otherwise stated.

[1]The BBC has produced and made available the BBC video sequences for use under the Creative Commons Attribution-NonCommercial 3.0 licence.

[2]These sequences and all intellectual property rights therein remain the property of Digiturk. These videos may be used according to Creative Commons Attribution-NonCommercial 3.0 Unported `http://creativecommons.org/licenses/by-nc/3.0/deed.en_US`. The dataset can be obtained from: `http://ultravideo.cs.tut.fi/`

[3]SJTU 4K Video Sequences: `http://medialab.sjtu.edu.cn/web4k/index.html`

[4]`http://www.ifp.illinois.edu/~dingliu2/iccv15/html/SRdemoFrame_set5.html`

[5]`http://www.ifp.illinois.edu/~dingliu2/iccv15/html/SRdemoFrame_set14.html`

- *Scale factor*: 2 and 4.
- *Patch size*: 5 with dictionary size 1024, and 10 with dictionary size 512. In our tests we set the patch size to 10 as default, unless otherwise stated.
- *Regularisation parameter* μ: 0.15.
- *Tolerance*: 0.05.

Following previous works, for our video SR experiments, we only consider the luminance channel in YCbCr colour space, as humans are more sensitive to luminance changes. The chroma components of the original video are interpolated using plain Bicubic interpolation. The evaluations for the Kato *et al.* [84], and the Yang *et al.* [168] models are calculated based on the MATLAB code and models provided by their respective authors.

7.4.1 Qualitative evaluation

We later demonstrate that our method is able to obtain high image quality metric values, however, the final judge for the image quality is the human viewer. It has been observed that although some methods generate visually appealing images, their Peak Signal-to-Noise (PSNR) values could be subjectively lower. Hence, the PSNR alone is not a reliable criterion for visual image quality. It must be noted that we did not perform a subjective perception test with many human subjects, and the reported qualitative results are based on the subjective opinion of the author only.

7.4.1.1 Video SR

To make a visual comparison between our model and other sparse-based methods, we super-resolve a GOP of 8 frames (the first 8 frames of a video) from all our test videos. We then compare the middle frame of the GOP with the corresponding SR image obtained

Figure 7.1: A GOP of 8 frames in Jockey, ShakeNDry, and Vehicles sequences up-sampled with upscaling factor 4 (480×270 to 1080p) with the VSRGOP + BP. Please refer to the supplementary material (available online `https://goo.gl/SKkG9V`) for full-size images.

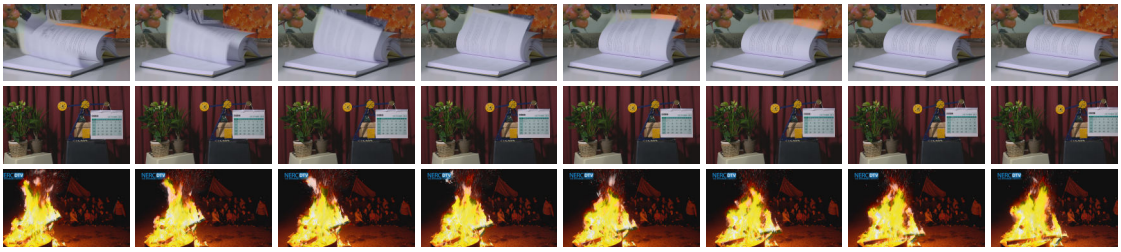


Figure 7.2: A GOP of 8 frames in Book, CalendarAndPlants, and CampfireParty sequences up-sampled with upscaling factor 3 (1080p to 4K UHD) with the VSRGOP + BP. Please refer to the supplementary material (available online `https://goo.gl/SKkG9V`) for full-size images.

by other algorithms. You can see the results of super-resolving a GOP of 8 frames from 480×270 to 1080p with an upscaling factor 4 in Figure 7.1, as well as the results for super-resolving from 1080p to 4K UHD with an upscaling factor 2 in Figure 7.2. Our algorithm is able to handle camera-induced motion in the background of the sequence well.

In Figure 7.3 we demonstrate a comparison between our method and four other methods. Here, a sequence has been super-resolved from 480×270 to 1080p with an upscaling factor 4. A cropped region of the image is shown that contains edges of printed fonts, as well as smooth texture and shading. While VSRGOP obtains fairer results than Bicubic and Yang [168], our method plus the Back-Projection (VSRGOP + BP) obtains higher visual reconstruction as well as better PSNR. The results in Kato + BP [84] tend to have grid-like and jagged artifacts.

Figure 7.3: Qualitative comparison for up-sampling the frame 2 of Vehicles sequence from 480×270 to 1080p using different methods. Please refer to the supplementary material for full images. For each sequence a crop of the image, as well as its respective full-image PSNR is shown.

Figure 7.4 shows another example with detailed soil texture and fine edges of objects such as the continuous track of the caterpillar excavator. Again, it can be observed that the VSRGOP + BP method produces more visually appealing results, with finer detail and better SR reconstruction.

Figure 7.5 shows more results for super-resolving sequences from 480×270 resolution to 1080p. In general our method is able to produce better texture, edge, and smooth-shaded region definitions for all the test videos; yet at the same time, the PSNR values of our results are the highest among competitors. While Bicubic interpolation produces overly smooth and watercolour-like images, our VSRGOP + BP is able to recreate

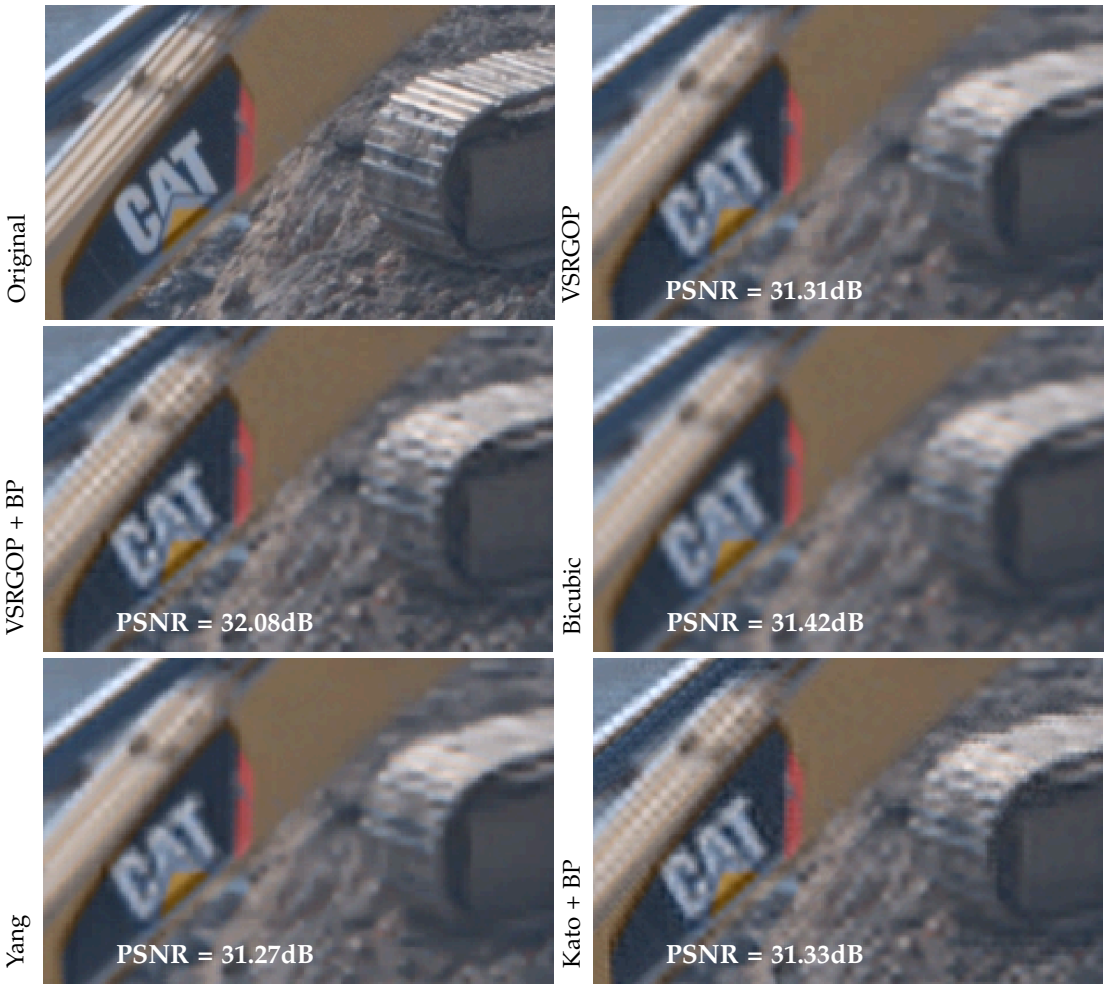


Figure 7.4: Qualitative comparison for up-sampling the frame 2 of ConstructionField sequence from 480×270 to 1080p using different methods. Please refer to the supplementary material for full images. For each sequence a crop of the image, as well as its respective full-image PSNR is shown.

both high-frequency and low-frequency components in the images. Kato + BP [84] is able to hallucinate the high-frequency content very well, however, it fails to produce visually appealing results on smoother regions. Moreover, the ringing and jagged artifacts produced by Kato + BP can be seen in the first three examples (HoneyBee, Jockey, and ParkAndBuildings sequences).

We also demonstrate how our method is able to hallucinate UHD super-resolution videos. Figure 7.6 shows examples of videos super-resolved from 1080p to 4K UHD

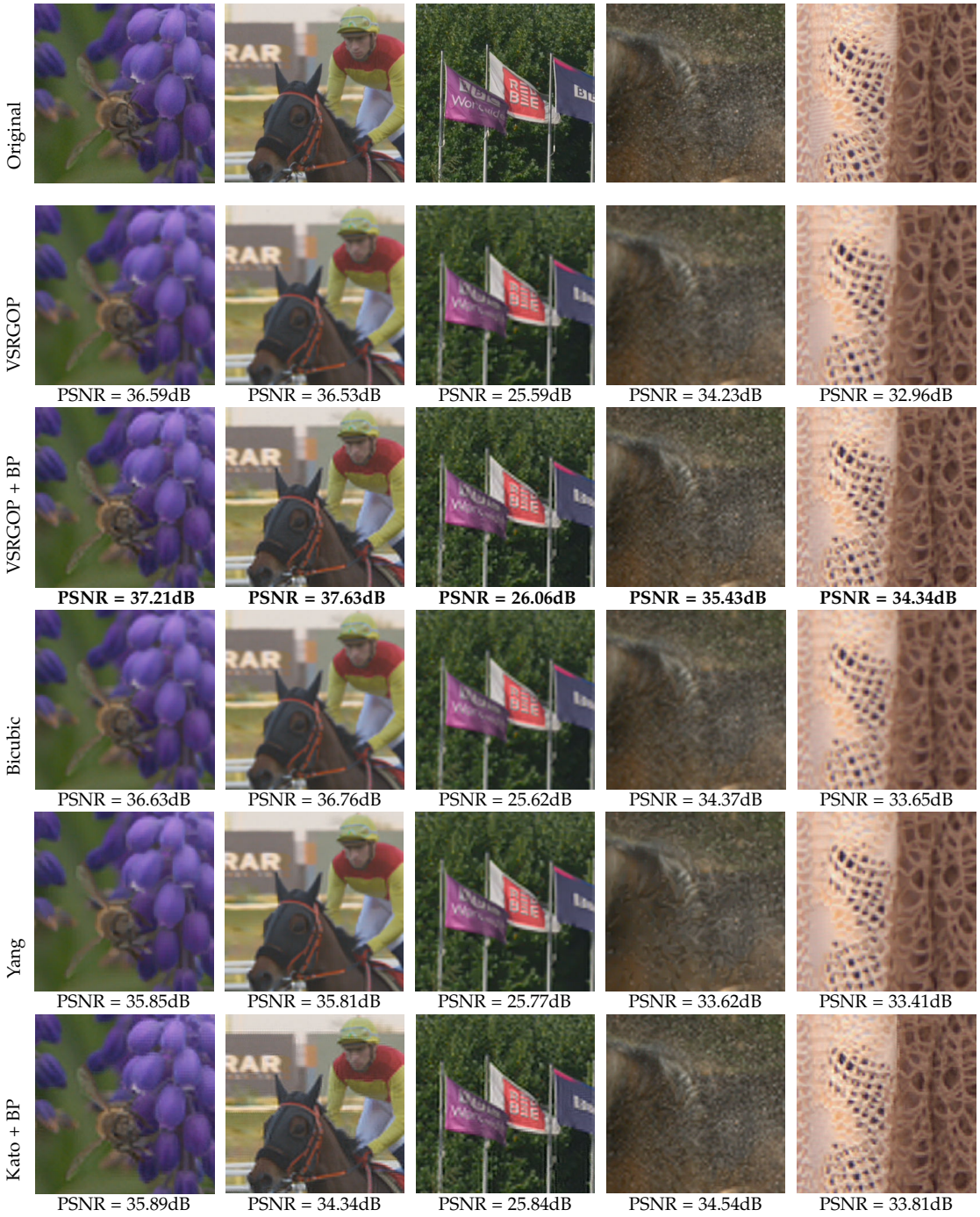


Figure 7.5: Qualitative comparison for up-sampling sequences from 480×270 to 1080p using different methods. Please refer to the supplementary material for full images. For each sequence a crop of the image, as well as its respective full-image PSNR is shown.

resolution. Since the resolution of the SR images is very high it is difficult to show fine details as well as larger portions of the image. Therefore, we cropped the image to small enough regions while still showing different parts of the produced HR frame. Similar to before, our method was able to produce better high frequency content, as well as well-defined edges and fine textures. Please see the supplementary material for full-sized images. Our method was able to produce more legible English and Chinese characters. For building facades our method was able to hallucinate more details both on the edges of the windows and on the windows with cast shadows and reflections. For the third example (CamfireParty) our method can hallucinate slightly sharper fire sparks.

Visually our VSRGOP + BP method produces better results in general with the subjective visual evaluation of the author. The obtained PSNR values for our multi-frame algorithm demonstrate superior performance as well. The advantage of using bilateral dictionaries compared with the unilateral dictionaries suggested by [84] is corroborated with our empirical results. Moreover, the visual results show that our multi-frame strategy outperforms the single-image algorithm in [168] and the multi-frame algorithm in [84]. As we will discuss later, the advantage of our method not only limited to higher qualitative performance, but also it achieves this with significant reduction of computational cost.

7.4.1.2 Single-Image SR

For our single-image SR tests, we regard the 3 RGB channels in the test image as 3 frames that can comprise a GOP. While the RGB channels could be super-resolved individually by our method, we choose to form a GOP structure, as this will remarkably reduce the computational cost. As there is very little textural change between the RGB channels, the obtained background frame for the 3 frames from the LRSD contains almost all of the information needed for the SR algorithm. The foreground part contains only the high-frequency content such as the edges or very fine textural boundaries in each of the RGB channels. For the other methods we super-resolve each of the RGB channels

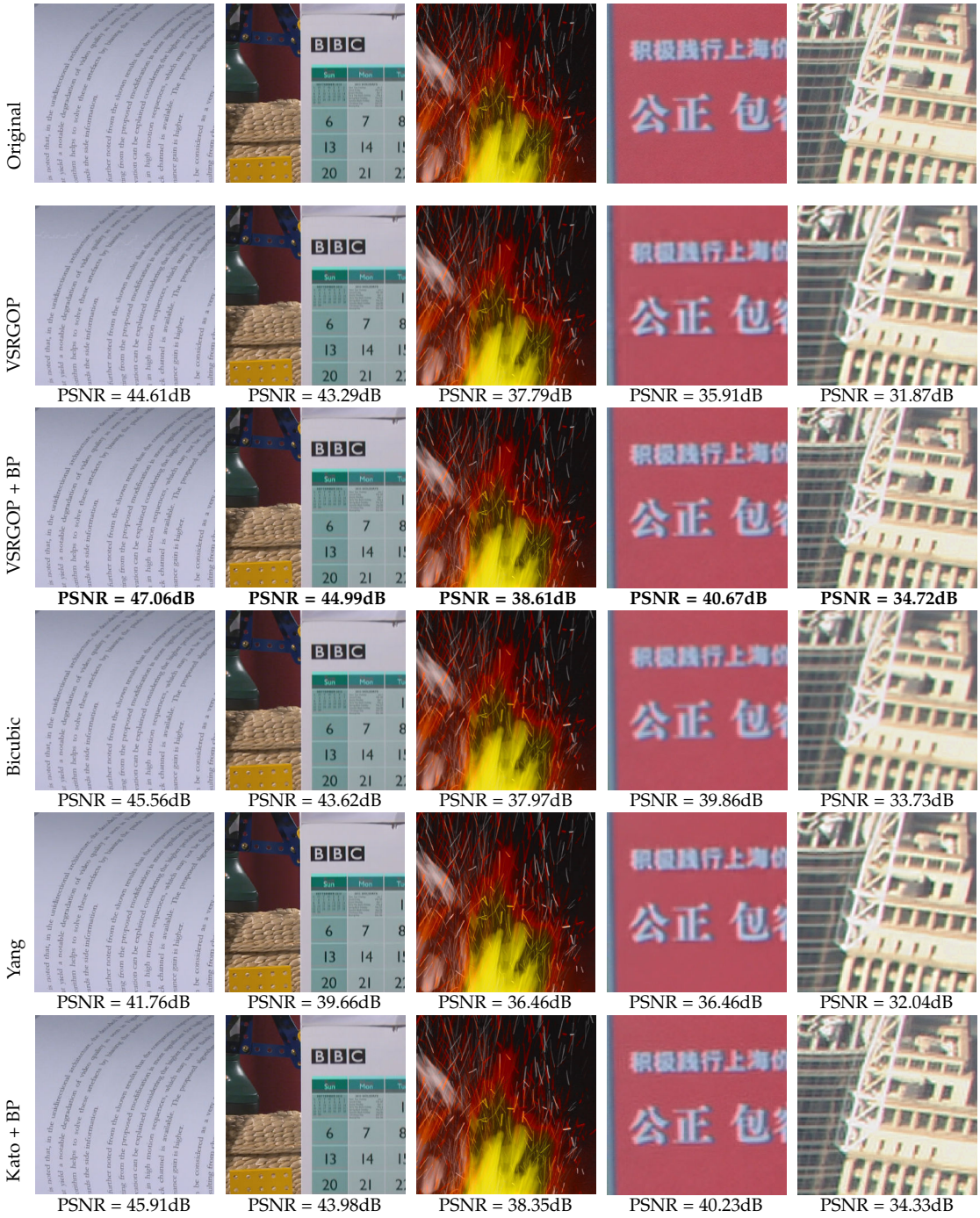


Figure 7.6: Qualitative comparison for up-sampling sequences from 1080p to 4K UHD using different methods. Please refer to the supplementary material for full images. For each sequence a crop of the image, as well as its respective full-image PSNR is shown.

individually. As one would expect, our method's SISR performs much faster than that of the other algorithms described in this section. In Figure 7.7 we show examples from the Set5 and Set14 datasets where an upscaling factor of 3 is used. In all the examples Kato + BP [84] is able to produce remarkably good HR images. Our VSRGOP + BP is able to produce more visually-appealing textures, despite the fact that by nature it is a video SR algorithm. Figure 7.8 shows more examples from the Set5 and Set14 datasets, where an upscaling factor of 4 is used. As the scale factor gets larger, Kato + BP [84] suffers from sharp and jagged artifacts, while Yang [168] misses some fine edge definitions. Our method nonetheless, produces superior results both visually (with the subjective visual evaluation of the author) and quantitatively.

7.4.2 Quantitative evaluation

In this section we analyse the proposed method's performance with PSNR image quality metric. Also, we compare the time consumption of our algorithm against state-of-the-art sparse-based SR methods.

7.4.2.1 Multi-Frame Video SR

Table 7-A shows the mean PSNR values for super-resolving all the frames in each of our test sequences individually. On average our algorithm wins for the SR problem. Our method provides between 0.77dB to 3.72dB improvement over its sparse-based predecessor, and between 0.52dB to 0.81dB improvement over the state-of-the-art sparse-based SR method. In Table 7-B we show an average time consumption comparison between our method and its predecessor sparse-based method [168] and state-of-the-art sparse-based method [84], for processing a 600-frame sequence. Our method is between 1.3× to 1.6× faster than its sparse-based predecessor and 271.1× to 424.6× faster than the state-of-the-art sparse-based SR method.

Recently, deep learning algorithms have had a great success in the SR problem. We

Table 7-A: Mean PSNR for up-sampling from 1080p to 4K UHD with upscaling factor 2, and from 480×270 to 1080p with upscaling factor 4 for all the frames in the sequences of 3 datasets. Our method provides between 0.77dB to 3.72dB improvement over its sparse-based predecessor, and between 0.52dB to 0.81dB improvement over the state-of-the-art sparse-based SR method.

	1080p to 4K UHD						480×270 to 1080p					
	VSRGOP	VSRGOP + BP	Kato [84]	Kato + BP [84]	Yang [168]	Bicubic	VSRGOP	VSRGOP + BP	Kato [84]	Kato + BP [84]	Yang [168]	Bicubic
Beauty	35.14	35.55	30.45	**36.65**	34.13	35.36	36.44	**36.77**	30.34	36.32	36.09	36.53
Book	44.98	46.82	35.19	**47.11**	41.73	46.04	36.77	**37.50**	29.32	35.81	36.49	36.89
Bosphorus	40.08	45.21	37.81	44.13	40.80	**45.93**	35.34	**36.92**	27.50	34.36	35.71	36.40
BundNightscape	40.84	**42.18**	31.36	39.15	37.63	41.19	31.80	**32.52**	25.31	31.26	31.76	31.82
CalendarAndPlants	43.36	**44.88**	35.06	43.78	39.53	43.72	33.14	33.87	28.13	**33.91**	33.15	33.17
CampfierParty	40.40	**41.49**	29.47	37.77	37.84	40.82	34.06	**34.80**	26.80	32.79	33.93	34.15
ConstructionField	36.68	**40.10**	32.26	39.23	36.71	40.02	31.22	**32.03**	24.68	31.24	31.28	31.42
Fountains	37.34	**38.74**	30.84	38.22	35.48	37.57	32.19	**32.83**	25.81	31.11	32.06	32.24
HoneyBee	38.08	38.23	34.70	**40.84**	36.33	38.15	36.67	**37.28**	27.48	35.91	35.91	36.71
Jockey	36.60	38.98	33.42	**41.39**	37.17	39.02	36.20	**37.96**	27.07	34.23	37.02	37.85
Library	36.72	**40.60**	31.57	38.80	37.09	39.70	29.81	30.40	25.25	**31.16**	29.95	29.74
Marathon	36.88	**37.58**	30.44	36.79	34.60	37.06	32.22	**33.04**	24.99	30.75	32.18	32.25
MenAndPlants	43.96	**45.64**	35.83	45.59	39.63	44.51	33.97	**34.81**	27.95	34.47	33.93	34.01
ParkAndBuildings	32.60	**37.50**	29.03	37.12	33.37	36.14	24.98	25.41	23.46	**28.41**	25.11	24.99
ReadySteadyGo	35.27	38.38	31.21	**39.68**	35.90	38.68	30.57	**31.57**	23.43	29.68	30.96	30.94
ResidentialBuilding	33.96	**38.51**	28.53	36.39	34.35	37.55	26.75	27.37	21.57	**27.62**	26.90	26.77
Runners	36.12	**37.50**	27.29	36.25	33.54	36.52	30.67	**31.41**	21.55	27.75	30.63	30.69
RushHour	41.70	**42.45**	33.07	41.22	38.03	42.05	33.61	**34.54**	25.12	33.02	33.46	33.65
Scarf	35.33	40.21	34.62	**41.68**	36.09	39.44	28.83	29.86	23.18	**30.48**	29.27	29.23
ShakeNDry	39.11	39.48	34.00	**40.67**	36.80	39.37	35.95	**36.87**	24.37	33.13	35.16	36.03
TallBuildings	32.03	**35.06**	26.80	34.59	32.21	33.99	25.81	26.23	22.25	**28.08**	25.89	25.86
TrafficAndBuilding	36.21	**40.49**	31.41	39.80	36.32	39.56	29.01	29.52	23.62	**29.64**	29.05	28.96
TrafficFlow	39.52	**40.37**	31.69	39.38	36.47	39.72	33.26	**34.15**	25.28	32.24	33.17	33.29
TreeShade	36.40	**38.02**	28.98	37.24	34.51	36.72	31.24	**31.90**	23.71	29.73	31.17	31.27
Vehicles	31.65	**35.69**	27.83	35.35	31.76	33.86	25.14	25.47	23.09	**28.51**	25.17	25.12
Wood	31.64	**36.94**	26.03	34.35	32.19	35.53	24.45	24.98	21.40	**27.16**	24.56	24.44
YachtRide	37.51	42.11	34.46	41.33	38.07	**42.60**	31.57	**32.56**	26.62	31.99	31.82	32.01
mean	37.41	**39.95**	31.61	39.43	36.23	39.29	31.54	**32.32**	25.16	31.51	31.55	31.72

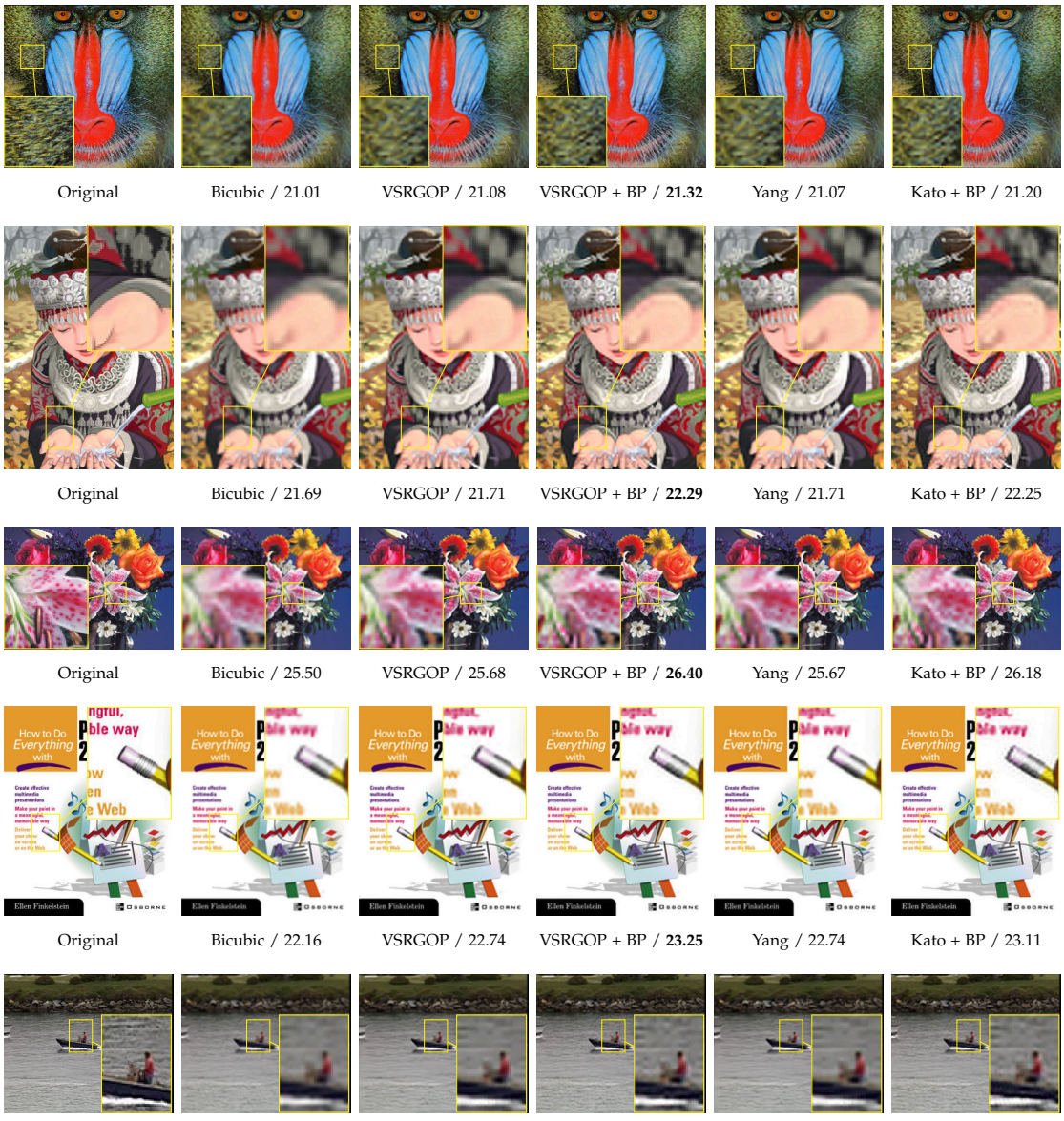


Figure 7.7: Single image super-resolution examples for "Baboon", "Comic", "Flowers", "PPT3", and "Coastguard" from Set5 and Set14 datasets with an upscaling factor of 3. PSNR values are shown under each sub-figure.

have selected the best published method SRCNN [31] with the 9-5-5 architecture trained on ImageNet dataset, and report its results in Table 7-C. Here an upscaling factor 4 is used. Our method outperforms SRCNN by 2.18dB on average, but SRCNN outperforms our method in 3 out of 7 categories. Moreover, the advantage of deep learning based methods is that they can be used in real-time processing. Although for applications

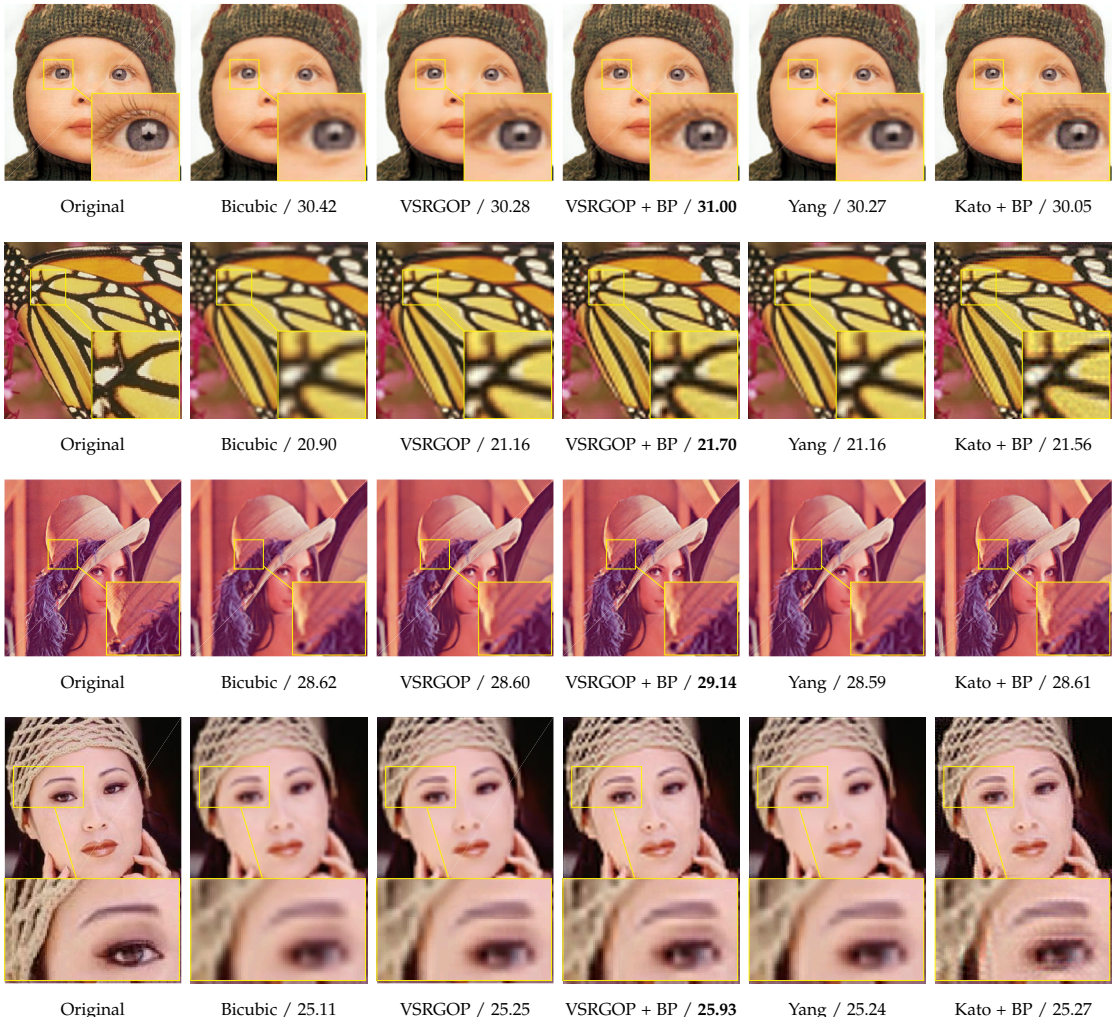


Figure 7.8: Single image super-resolution examples for "Baby", "Butterfly", "Lena", and "Woman" from Set5 and Set14 datasets with an upscaling factor of 4. PSNR values are shown under each sub-figure.

where exact reconstruction is vitally important our method offers a potential to be a good alternative.

7.4.2.2 Effect of Patch Size

The experimental results reported in the previous sections show that the LRSD, and the sparsity prior for image patches is very effective in regularising the ill-posed problem of SR. As mentioned, we fix the patch sizes to 5 and 10, for the dictionary sizes of 1024

Table 7-B: Average time consumption comparison between our method and its predecessor sparse-based method [168] and state-of-the-art sparse-based method [84], for processing 1 frame. Our method is between 1.3× to 1.6× faster than its sparse-based predecessor and 271.1× to 424.6× faster than the state-of-the-art sparse-based SR method.

1080p to 4K UHD

	VSRGOP + BP	Yang [168]	Kato + BP [84]
time (h:mm:ss.s)	**0:08:20.9**	0:10:32.4	58:57:5.5

480×270 to 1080p

	VSRGOP + BP	Yang [168]	Kato + BP [84]
time (h:mm:ss.s)	**0:01:32.9**	0:02:30.6	6:59:43.0

Table 7-C: Comparison with state-of-the-art Super-Resolution method with a Deep Learning approach SRCNN 9-5-5 [31] trained on ImageNet dataset, using an upscaling factor 4 in terms of PSNR.

	SRCNN [137]	VSRGOP + BP
Bosphorus	37.53	**45.21**
ReadySetGo	33.69	**38.38**
Beauty	**39.48**	35.55
YachtRide	33.17	**42.11**
ShakeNDry	36.68	**39.48**
HoneyBee	**40.51**	38.23
Jockey	**41.55**	38.98
mean	37.52	**39.70**

and 512 respectively. Intuitively, larger dictionary size should be able to describe more variation in the data, and as such yield better approximation. Also, larger dictionary size would be more computationally expensive [168]. We therefore, remedy this by using a smaller patch size for the larger dictionary and a larger patch size for the smaller dictionary, and expect to obtain similar results for both of these. Table 7-D shows a comparison between the two patch sizes. We show the results for a GOP of 8 frames, averaged across all our datasets. We have used the SVD variant of our algorithm. In our method by default we select a smaller dictionary size, while increasing the patch size to allow for enough expressive power. A larger patch size with a smaller dictionary achieves 1 dB better reconstruction quality, while providing ∼11× faster processing.

Table 7-D: Effect of patch size: Two patch sizes 5 (dictionary size 1024) and 10 (dictionary size 512) are used to process a GOP of 8 frames. A larger patch size used in combination with a smaller dictionary would speed up the process by ∼11×, yet also increases the quality by ∼1dB. Bicubic method shown here is for baseline performance analytics.

		1080p to 4K UHD		480×270 to 1080p	
		5	10	5	10
VSRGOP	mean PSNR	41.56	**42.12**	35.65	**37.27**
VSRGOP + BP	mean PSNR	43.69	**44.26**	36.07	**37.21**
time (h:mm:ss.s)		7:42:21.7	**0:43:14.8**	1:31:58.6	**0:08:05.8**
Bicubic	mean PSNR	39.08		30.73	

Table 7-E: Effect of SVD: the results for processing a GOP of 8 frames are shown. When using the SVD-free algorithm, the quality degrades between 0.16 to 0.69dB, but the time consumption is reduced by 6% to 18%. Bicubic method shown here is for baseline performance analytics.

		1080p to 4K UHD		480×270 to 1080p	
		SVD	SVD-Free	SVD	SVD-Free
VSRGOP	mean PSNR	**42.12**	41.96	**37.21**	36.52
VSRGOP + BP	mean PSNR	**44.26**	44.06	**37.27**	36.62
time (h:mm:ss.s)		0:33:14.5	**0:31:13.1**	0:07:26.8	**0:06:18.5**
Bicubic	mean PSNR	39.08		30.73	

7.4.2.3 Effect of SVD

In this section we analyse the SVD-free variant of our algorithm. In Table 7-E the results for processing a GOP of 8 frames are shown. The patch size 10 has been used here, with a dictionary size of 512. When the SVD-free algorithm is used, the quality degrades between 0.16 to 0.69dB, but the time consumption is reduced by 6% to 18%.

7.4.2.4 Effect of GOP Size

Here the effect of GOP size is analysed. For this test we super-resolve our sequences with five different GOP sizes of 8, 16, 24, 32, and 64. We use the SVD variant of our method, along with a patch size 10, and dictionary size 512. In Figure 7.9 the results for upscaling with factors 2 and 4 (shown in parentheses next to each legend) are demonstrated. It can be seen that as the GOP size is increased the time consumption increases too, with

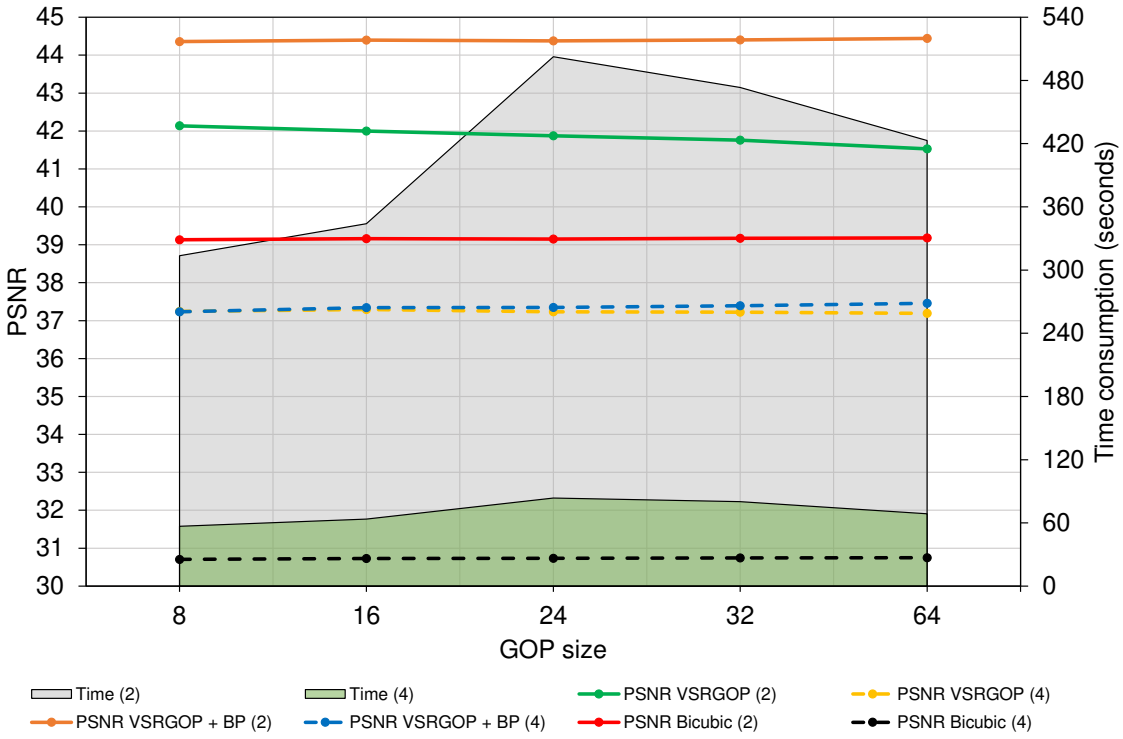


Figure 7.9: Effect of GOP size on PSNR and time consumption for processing 1 frame. Five GOP sizes 8, 16, 24, 32, and 64 are used. The time consumption increases with GOP size, with it being the highest at GOP size 24, followed by 32 and 64. The PSNR remains robustly unchanged as the GOP size is altered. Following this, we use GOP size 8 for our tests while we can safely assume that it will give us the maximal quality, while providing the least time consumption. Upscaling factors of 2 and 4 are used and shown in parentheses next to each legend.

the GOP size 24 being the most expensive setting (the gray and green filled-in curves). However, the PSNR remains robustly unchanged (the horizontal solid and dashed lines), with only small fluctuations.

7.5 Conclusions

In this chapter we introduced a new sparsity-based video super-resolution method, that exploits the spatio-temporal information of the video sequence by a low-rank and sparse decomposition algorithm. Our method builds upon sparse representations in terms of

coupled dictionaries jointly trained from high- and low-resolution image patch pairs. Our low-rank and sparse decomposition provides significant reductions in computation cost, while increasing the visual and quantitative quality of the reconstruction results by exploiting the spatio-temporal information that can be shared among adjacent frames of a video. Extensive experimental evaluation on 3 video datasets indicate the efficacy and effectiveness of the proposed algorithm in video super-resolution for HD and UHD content. Furthermore, we demonstrated the efficacy of our method for the single-image super-resolution problem, and showed that it can be successfully applied to single images, yet at the same time providing better reconstruction quality as well as less computation time.

Chapter 8

Conclusions and Future Development

8.1 Summary

In this thesis we have presented a novel Approximated Robust Principal Component Analysis method and validated its efficacy and effectiveness in several computer vision applications. Our method which builds upon the existing RPCA model has a four-term decomposition, namely a low-rank component, a sparse component, additive noise, and global background motion transformation components. The immediate advantages of our proposal are: tunable rank of the low-rank component for specific problems that is obtained by a further relaxation of RPCA with respect to the rank of the low-rank part; new structured-sparsity inducing norms that can better describe the spatial connectivity of the pixels in the sparse component corresponding to the foreground objects in the scene; better initialisation strategies for the Approximated RPCA that result in faster convergence as well as better approximation of the correct solution; the ability to apply the RPCA problem to the cases where image data is captured with a moving camera; robustness to noise and corruptions; more computationally scalable solutions

with dimensionality reduction algorithms such as deterministic and randomised Column Subset Selection Problem; a SVD-free solution to the rank-minimisation problem in case a rank-1 low-rank component is sought; incremental decomposition for long video sequences; and lastly, computationally cheaper algorithms for solving ARPCA.

In Chapter 2, the main subspace estimation concepts were described, with particular focus on the low-rank and sparse decomposition methods via the RPCA. A compact summary of the performance of the fundamental algorithms to solve RPCA was provided, and an overview of their limitations was discussed.

In Chapter 3, an adaptation of the Approximated RPCA for the HEVC standard for video compression was presented, that enables higher bitrate savings for the guaranteed reconstruction quality. The LRSD adaptation to HEVC has been made possible by a novel incremental decomposition with many configurations made available for HEVC requirements. A new norm-minimisation for this application was proposed that works in the same quadtree coding units as HEVC, as well as modified low-rank approximation for low-bitrate background encoding. The whole LRSD-HEVC framework has been extended to handle sequences with camera-induced motion that are the most difficult cases in video compression. Optimal parameter selection has been an area of interest in the proposed method, and as such we have left enough room for reconfigurability of our model to achieve the best trade-off when used in conjunction with an HEVC encoder/decoder.

In Chapter 4, an Approximated RPCA was presented for robust alignment and recovery of corrupted linearly correlated images and videos. We also suggested applications such as batch image alignment, recovery of face images form corrupted data for face recognition, video stabilisation, image mosaicking, inpainting etc. Our proposed formulation directly impacts the speed of convergence of the algorithm, making it suited for real-time applications, as well as handling of larger misalignments compared to the contenders.

In Chapter 5, we presented a novel background subtraction method and validated

its efficacy and effectiveness with extensive testing. The method is based on an existing model, namely RPCA, but with new sparsity-inducing norms and group-structured sparsity constraints. Whilst our simple DBSS model produces crisp and well-defined genuine foreground segmentation, our more elaborate DSPSS model surpasses its performance by taking advantage of the natural shape and structure of objects in the scene. Both our sparsity models dynamically evolve to best describe genuine foreground objects in the scene, which gives them a significant advantage when it comes to handling dynamic backgrounds, or foreground aperture. To make the problem computationally scalable we proposed using deterministic and randomised CSSP for low-rank matrix estimation. Moreover, a novel tandem initialisation method is proposed to speed up convergence and remove ghosting effects persisting in RPCA-based methods. Specifically, our model is able to learn a robust background model that can change over time, to cope with a variety of scene changes, in comparison with the existing more heuristic RPCA-based methods. It proves itself to have excellent performance in dealing with heavy noise, thanks to the approximated RPCA model where the residual Gaussian noise is discarded into a third matrix in the decomposition. In addition, estimation of background motion induced by a jittering or moving camera is performed simultaneously with low-rank approximation, that results in excellent performance in shaky videos.

In Chapter 6, we addressed the problem of subspace clustering. Given a set of data samples approximately drawn from a union of multiple subspaces, our goal is to cluster the samples into respective subspaces, and also remove possible outliers. We propose a novel Approximated Robust PCA Clustering (ARPCAC) method, that seeks the lowest rank representation among all the candidates that can represent the samples drawn from camera-induced motion. The proposed method involves extracting the point trajectories only induced by object motion, from the pool of all motions with our ARPCAC method, and then projecting them onto a 5-dimensional space, using PowerFactorisation. We apply our algorithm to the problem of segmenting multiple motions in video and furthermore, we extend our work to the problem of face clustering. Conducted experiments

show that our approach significantly outperforms state-of-the-art methods.

In Chapter 7, we introduced a new sparsity-based video super-resolution method, that exploits the spatio-temporal information of the video sequence by a low-rank and sparse decomposition algorithm. Our method builds upon sparse representations in terms of coupled dictionaries jointly trained from high- and low-resolution image patch pairs. Our low-rank and sparse decomposition provides significant reductions in computation cost, while increasing the visual and quantitative quality of the reconstruction results by exploiting the spatio-temporal information that can be shared among adjacent frames of a video. Extensive experimental evaluation on 3 video datasets indicate the efficacy and effectiveness of the proposed algorithm in video super-resolution for HD and UHD content. Furthermore, we demonstrated the efficacy of our method for the single-image super-resolution problem, and showed that it can be successfully applied to single images, yet at the same time providing better reconstruction quality as well as less computation time.

8.2 Key Contributions

The major contributions in this thesis are detailed in this section.

Approximated RPCA for HEVC

- An incremental GOP-based decomposition was presented.
- Configurability of the LRSD output for HEVC, namely with, variable GOP size, variable block size, variable number of backgrounds per GOPs, variable quadtree division, and low-bitrate background generation by mutual CU position estimation between background and foregrounds of a GOP.

Approximated RPCA for alignment and recovery of corrupted linearly correlated images and video frames

- A novel Approximated Robust Principal Component Analysis framework for recovery of a set of linearly correlated images, with applications in batch image alignment, recovery of face images form corrupted data for face recognition, video stabilisation, image mosaicking, and image inpainting.

- Decomposition of a batch of realistic, unaligned, and corrupted images as a sum of a low-rank and a sparse corruption matrix, while simultaneously aligning the images according to the optimal image transformations.

Approximated RPCA for backgorund modelling and foreground segmentation

- The decomposition into three terms, namely a low-rank, a sparse, and a Gaussian i.i.d. noise part for discarding false positive alarms was proposed.

- A novel dynamic tree-structured sparsity inducing norm was proposed and realised, as well as a dynamic block structure, and a dynamic superpixel structure for the group sparsity.

- A tandem algorithm for removal of unwanted ghosting effects that persist in background subtraction, and targets unascertained prior knowledge of distribution of outliers was presented.

- A dimensionality reduction for RPCA problem via the column subset selection algorithm that eliminates the bootstrapping problem, and reduces computational complexity and cost was studied.

Subspace clustering

- An Approximated Robust Principal Component Analysis Clustering (ARPCAC) method was proposed that can cluster the samples into respective subspaces, and also remove possible outliers, while revealing each subspace's motion.

- APRCAC was also shown to be effective in the problem of face clustering.

UHD video super-resolution

- A novel sparse-based algorithm for multi-frame video super-resolution (SR) was proposed, that was shown to also be applicable to the problem of single-image SR.
- Our Video Super-Resolution in Group of Pictures (VSRGOP) method incorporates the spatio-temporal information in videos, by a low-rank plus sparse decomposition of the video sequence.
- A Group of Pictures (GOP) structure was used, where a rank-1 low-rank component is sought from the video sequence, that recovers the shared spatio-temporal information among the frames in the GOP. Then the obtained low-rank frame and the sparse frames are super-resolved separately by our sparse coding mechanism.
- Our proposed method obtains significant time reductions for calculating a SR video in the sparse coding framework, while increasing the visual and quantitative quality of the reconstruction results.

8.3 Future Work

A number of improvements to our Approximated RPCA model can be considered that can benefit the tasks of HEVC, batch image alignment, background modelling and foreground segmentation, subspace clustering, and video super-resolution.

For HEVC applications the decomposition could be adapted to work on 3D matrices, i.e., a tensor representation for video content, as opposed to vectorised representation of frames of the video as columns of a 2D matrix. Moreover, it would be beneficial if each coding unit could be decomposed individually into its low-rank, sparse, and noise

parts along with a motion estimation for the coding unit. This would make the problem very complex, and therefore a mechanism to handle the high computational complexity must be devised. Furthermore, a series of thorough subjective human perception tests must be performed to clearly analyse various aspects of the designed method. Another important aspect that must be studied is to index the proposed method's performance in terms of quantisation parameter (QP) which is an established quantity vs. quality metric in HEVC. QP can be specified to control bitrate or the quality, and can take 52 values from 0 to 51 for 8-bit video sequences. If the QP is low, the bitrate will increase and the quality will improve, and vice versa. It would also be beneficial to explore other incremental decomposition methods that might speed up the decomposition process. Optimisation becomes of vital importance for the HEVC applications, as the resolution of the input video sequences increases. The main computational cost of our method lies in the low-rank estimation, and although we alleviated this with our single background per GOP solution, more effective optimisation methods are still needed to satisfy the needs of HEVC.

For the batch image alignment and recovery, one of the most important questions is how to extend our framework to more general classes of transformations such as non-rigid and non-parametric that are exhibited in general video data, while providing the same practical guarantees for the amount of misalignment and corruption it can handle. Another important aspect is to address the scalability issue with this model; unlike other chapters in this thesis, in the batch image alignment and recovery, the low-rank part that is yielded by the decomposition is more interesting to us than the sparse part. Therefore, the optimisation techniques used in other chapters, such as incremental decomposition, or the CSSP cannot be used for the problem at hand.

Our model is yet another batch method, as the frames need to be stored for obtaining a background model; although we alleviated this limitation to some extent by the CSSP in Chapter 5, further optimisation is required to achieve real-time performances. This could include a learning stage followed by incremental updates as the frames arrive. In

another attempt to further optimise the low-rank estimation, an initial low-rank component from the first few frames of a video sequence could be calculated, followed by incremental rank-1 updates on the low-rank component with each arriving frame. This includes additive modifications of a SVD to reflect updates and edits of the input data matrix as the frames arrive. Spatio-temporal constraints are also another area of attraction for our method. For background modelling, sudden illumination changes are slowly adapted by the background model, and hence it fails to handle some indoor lighting changes. The low-rank estimation indirectly encapsulates temporal information, however a more explicit temporal solution such as tensor decomposition could benefit our method. Furthermore, a more sophisticated model should be able to handle shadows, that are not interesting for later processing. Solutions to these problems could be adapted to our method. Our method currently is able to provide a binary classification of pixels, while more recently many methods have emerged that address the foreground/background segmentation problem as a multi-class segmentation which is specifically useful for semantic segmentation. Our method could be extended to decompositions with more than one low-rank or sparse component, by possibly exploiting the motion trajectories of each low-rank or sparse subspace.

For the problem of subspace clustering we would like to extend ARPCAC to not rely on PowerFactorisation and Spectral Clustering for the separation of the extracted independent motions, by the low-rank and sparse decomposition. Also, the current ARPCAC model can only decompose the pool of existing motions into two components, the global background motion trajectories, and object-induced trajectories. A more sophisticated model must be able to decompose motions into more than 2 clusters.

Another interesting topic that is worth exploring is extending our VSRGOP + BP model for UHD video SR into a deep learning framework to achieve real-time performances, as slow speed is a major drawback of our method. The adaptation of our method into a deep learning model could be achieved by encouraging the neurons of certain layers in a convoluational neural network to learn sparse and low-rank repre-

sentations of the data. Also, since convolutional neural networks do not offer temporal consistency, incorporation of a low-rank representation that encapsulates temporal information into the network, seems to offer an excellent research potential. In future work, we would also like to extend our work to the problem of upsampling medical images. Our method is indeed slower than current deep learning-based models, while performing on a par with one of the most prominent ones as demonstrated in Chapter 7. Nevertheless, we would need to analyse the performance of our model when the modality of the images changes. Also in further studies, our method should be analysed against how much it can cope with various compression artefacts as well as motion blur. Since PSNR may not be the best quality metric, other quality metrics such as structural similarity (SSIM) index must be employed to provide more insight into the results. Finally, to provide more credible qualitative evaluations, we need to perform a series of thorough human subjective perception tests. There are various methods for achieving this, but the most efficient way could be realised by running Amazon Turk tasks where two images are shown side-by-side to the human subjects, for them to specify which they prefer visually.

Publications

Erfanian Ebadi S., Guerra Ones V., and Izquierdo E. (2017), "UHD Video Super-Resolution using Aided Sparse Representation", submitted to IEEE Transactions on Image Processing, (TIP).

Erfanian Ebadi S., Guerra Ones V., and Izquierdo E. (2017), "UHD Video Super-Resolution using Low-Rank and Sparse Decomposition", International Conference on Computer Vision, Workshop on Robust Subspace Learning and Applications in Computer Vision, (ICCV RSL-CV), Venice, Italy.

Erfanian Ebadi S., and Izquierdo E. (2017), "Multiple Subspaces Separation in Case of Camera Motion", IET International Conference on Imaging for Crime Detection and Prevention, (IET ICDP), Madrid, Spain.

Erfanian Ebadi S., and Izquierdo E. (2017), "Foreground Detection with Dynamic Tree-Structured Sparse RPCA", IEEE Transactions on Pattern Analysis and Machine Intelligence, (TPAMI).

Erfanian Ebadi S., and Izquierdo E. (2016), "Foreground Segmentation via Dynamic Tree-Structured Sparse RPCA", IEEE European Conference on Computer Vision, (ECCV), Amsterdam, Netherlands.

Erfanian Ebadi S., Guerra Ones V., and Izquierdo E. (2016), "Dynamic Tree Structured Sparse RPCA via Column Subset Selection for Background Modeling and Fore-

ground Detection", Image Processing, IEEE International Conference on Image Processing, (ICIP) Phoenix, AZ, USA.

Erfanian Ebadi S., Guerra Ones V., and Izquierdo E. (2015), "Approximated Robust Principal Component Analysis for Improved General Scene Background Subtraction", arXiv:1603.05875.

Erfanian Ebadi S., Guerra Ones V., and Izquierdo E. (2015), "Efficient Background Subtraction with Low-rank and Sparse Matrix Decomposition", IEEE International Conference on Image Processing, (ICIP), Québec City, Canada.

Erfanian Ebadi S., Izquierdo E. (2015), "Approximated RPCA for Fast and Efficient Recovery of Corrupted and Linearly Correlated Images and Video Frames", International Conference on Systems, Signals and Image Processing, (IWSSIP), London, UK.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 11, pp. 2274–2282, Nov 2012.

[2] Anonymous, "CDet," 2012. [Online]. Available: http://wordpress-jodoin.dmi.usherb.ca/method/146/

[3] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *arXiv preprint arXiv:1511.00561*, 2015.

[4] O. Barnich and M. V. Droogenbroeck, "ViBe: A universal background subtraction algorithm for video sequences," *IEEE Transactions on Image Processing*, pp. 1709–1724, 2011.

[5] O. Barnich and M. Van Droogenbroeck, "ViBe: A powerful random technique to estimate the background in video sequences," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on.* IEEE, 2009, pp. 945–948.

[6] R. Basri and D. W. Jacobs, "Lambertian reflectance and linear subspaces," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 2, pp. 218–233, 2003.

[7] S. Becker, E. Candès, and M. Grant, "TFOCS: Flexible first-order methods for stable principal component pursuit," *SIAM Conference on Optimization: Low-*

Rank Matrix Optimization Symposium, 2011.

[8] D. D. Bloisi, *Background Modeling and Foreground Detection for Video Surveillance*, T. Bouwmans, F. Porikli, B. Hrferlin, and A. Vacavant, Eds. CRC Press, Taylor and Francis Group, July 2014.

[9] C. Boutsidis, M. W. Mahoney, and P. Drineas, "An improved approximation algorithm for the column subset selection problem," in *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics, 2009, pp. 968–977.

[10] T. Bouwmans, "Recent advanced statistical background modeling for foreground detection: A systematic survey," *Recent Patents on Computer Science*, vol. 4, no. 3, 2011.

[11] T. Bouwmans and E. Zahzah, "Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance," *Special Isssue on Background Models Challenge , Computer Vision and Image Understanding*, 2014.

[12] T. Bouwmans, N. S. Aybat, and E.-h. Zahzah, *Handbook of Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing*. CRC Press, 2016.

[13] T. Bouwmans, A. Sobral, S. Javed, S. K. Jung, and E.-H. Zahzah, "Decomposition into low-rank plus additive matrices for background/foreground separation: A review for a comparative evaluation with a large-scale dataset," *Computer Science Review*, vol. 23, pp. 1–71, 2017.

[14] M. E. Broadbent, M. Brown, K. Penner, I. Ipsen, and R. Rehman, "Subset selection algorithms: Randomized vs. deterministic," *SIAM Undergraduate Research Online*, vol. 3, pp. 50–71, 2010.

[15] S. Brutzer, B. Höferlin, and G. Heidemann, "Evaluation of background subtraction techniques for video surveillance," in *Computer Vision and Pattern Recognition (CVPR) IEEE*.

[16] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. on Optimization*, vol. 20, no. 4, pp. 1956–1982, Mar. 2010. [Online]. Available: http://dx.doi.org/10.1137/080738970

[17] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, pp. 11:1–11:37, Jun. 2011.

[18] V. Cevher, M. F. Duarte, C. Hegde, and R. Baraniuk, "Sparse signal recovery using markov random fields," in *Advances in Neural Information Processing Systems*, 2009, pp. 257–264.

[19] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, "Rank-sparsity incoherence for matrix decomposition," *SIAM Journal on Optimization*, vol. 21, no. 2, pp. 572–596, 2011.

[20] R. Chartrand, "Non-convex splitting for regularized low-rank + sparse decomposition," *IEEE Transactions on Signal Processing*, vol. 60, pp. 5810–5819, 2012.

[21] G. Chen and G. Lerman, "Spectral curvature clustering SCC," *International Journal of Computer Vision*, vol. 81, no. 3, pp. 317–330, 2009.

[22] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *arXiv preprint arXiv:1606.00915*, 2016.

[23] Z. Chen, S. D. Babacan, R. Molina, and A. K. Katsaggelos, "Variational Bayesian methods for multimedia problems," *IEEE Transactions on Multimedia*, vol. 16, no. 4, pp. 1000–1017, 2014.

[24] S.-c. S. Cheung and C. Kamath, "Robust techniques for background subtraction in urban traffic video," vol. 5308, 2004, pp. 881–892. [Online]. Available: http://dx.doi.org/10.1117/12.526886

[25] M. Collins, S. Dasgupta, and R. E. Schapire, "A generalization of principal components analysis to the exponential family," in *Advances in neural information processing systems*, 2002, pp. 617–624.

[26] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[27] J. P. Costeira and T. Kanade, "A multibody factorization method for independently moving objects," *International Journal of Computer Vision*, vol. 29, no. 3,

pp. 159–179, 1998.

[28] D. Culibrk, O. Marques, D. Socek, H. Kalva, and B. Furht, "Neural network approach to background modeling for video object segmentation," *Neural Networks, IEEE Transactions on*, vol. 18, no. 6, pp. 1614–1627, 2007.

[29] Y. Deng, Q. Dai, R. Liu, Z. Zhang, and S. Hu, "Low-rank structure learning via non-convex heuristic recovery," *IEEE transactions on neural networks and learning systems*, vol. 24, no. 3, pp. 383–396, 2013.

[30] X. Ding, L. He, and L. Carin, "Bayesian robust principal component analysis," *IEEE Transactions on Image Processing*, vol. 20, no. 12, pp. 3419–3430, 2011.

[31] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[32] D. L. Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[33] ——, "For most large underdetermined systems of linear equations the minimal ℓ_1-norm solution is also the sparsest solution," *Communications on pure and applied mathematics*, vol. 59, no. 6, pp. 797–829, 2006.

[34] P. Drineas, M. W. Mahoney, and S. Muthukrishnan, "Relative-error CUR matrix decompositions," *SIAM Journal on Matrix Analysis and Applications*, vol. 30, no. 2, pp. 844–881, 2008.

[35] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani *et al.*, "Least angle regression," *The Annals of statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[36] Y. C. Eldar, P. Kuppinger, and H. Bölcskei, "Block-sparse signals: Uncertainty relations and efficient recovery," *Signal Processing, IEEE Transactions on*, vol. 58, no. 6, pp. 3042–3054, 2010.

[37] A. Elgammal, D. Harwood, and L. Davis, "Non-parametric model for background subtraction," in *European Conference on Computer Vision ECCV 2000*. Springer, 2000, pp. 751–767.

[38] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*,

vol. 35, no. 11, pp. 2765–2781, 2013.

[39] S. Erfanian Ebadi, , and E. Izquierdo, "Approximated RPCA for fast and efficient recovery of corrupted and linearly correlated images and video frames," in *Systems, Signals and Image Processing (IWSSIP), 2015 International Conference on*, Sept 2015, pp. 49–52.

[40] S. Erfanian Ebadi, V. Guerra Ones, and E. Izquierdo, "Efficient background subtraction with low-rank and sparse matrix decomposition," in *Image Processing (ICIP), 2015 IEEE International Conference on*, Sept 2015, pp. 4863–4867.

[41] ——, "Approximated robust principal component analysis for improved general scene background subtraction," *CoRR*, vol. abs/1603.05875, 2016. [Online]. Available: http://arxiv.org/abs/1603.05875

[42] ——, "Dynamic tree structured sparse RPCA via column subset selection for background modeling and foreground detection," in *Image Processing (ICIP), 2016 IEEE International Conference on*, 2016.

[43] ——, "UHD video super-resolution using aided sparse representation," *submitted to Image Processing (TIP), IEEE Transactions on*, 2017.

[44] ——, "UHD video super-resolution using low-rank and sparse decomposition," in *International Conference on Computer Vision Workshop on Robust Subspace Learning and Applications in Computer Vision, (ICCV RSL-CV), Venice, Italy*, 2017.

[45] S. Erfanian Ebadi and E. Izquierdo, *Foreground Segmentation via Dynamic Tree-Structured Sparse RPCA*. Springer International Publishing, 2016, pp. 314–329. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-46448-0_19

[46] ——, "Foreground segmentation via dynamic tree-structured sparse RPCA," in *European Conference on Computer Vision*. Springer International Publishing, 2016, pp. 314–329.

[47] ——, "Foreground detection with dynamic tree-structured sparse RPCA," *Pattern Analysis and Machine Intellignece (TPAMI), IEEE Transactions on*, 2017.

[48] ——, "Multiple subspaces separation in case of camera motion," in *IET International Conference on Imaging for Crime Detection and Prevention, (IET ICDP),*

Madrid, Spain, 2017.

[49] R. H. Evangelio, M. Pätzold, and T. Sikora, "Splitting gaussians in mixture models," in *Advanced Video and Signal-Based Surveillance (AVSS), 2012 IEEE Ninth International Conference on.* IEEE, 2012, pp. 300–305.

[50] R. H. Evangelio and T. Sikora, "Complementary background models for the detection of static and moving objects in crowded environments," in *Advanced Video and Signal-Based Surveillance (AVSS), 2011 8th IEEE International Conference on.* IEEE, 2011, pp. 71–76.

[51] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[52] P. Favaro, R. Vidal, and A. Ravichandran, "A closed form solution to robust subspace estimation and clustering," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on.* IEEE, 2011, pp. 1801–1807.

[53] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories," *Computer Vision and Image Understanding*, vol. 106, no. 1, pp. 59–70, 2007.

[54] J. Feng, Z. Lin, H. Xu, and S. Yan, "Robust subspace segmentation with block-diagonal prior," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 3818–3825.

[55] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[56] A. Frieze, R. Kannan, and S. Vempala, "Fast monte-carlo algorithms for finding low-rank approximations," *Journal of the ACM (JACM), volume=51, number=6, pages=1025–1041, year=2004, publisher=ACM.*

[57] D. Gabay and B. Mercier, "A dual algorithm for the solution of non-linear variational problems via finite element approximation," *Computers & Mathematics with Applications*, vol. 2, no. 1, pp. 17–40, 1976.

[58] Z. Gao, L.-F. Cheong, and Y.-X. Wang, "Block-sparse RPCA for salient motion detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 10, pp. 1975–1987, Oct 2014.

[59] R. Glowinski and A. Marroco, "Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non-linéaires," *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique*, vol. 9, no. R2, pp. 41–76, 1975.

[60] J. Goes, T. Zhang, R. Arora, and G. Lerman, "Robust stochastic principal component analysis," in *Artificial Intelligence and Statistics*, 2014, pp. 266–274.

[61] A. Goh and R. Vidal, "Segmenting motions of different types by unsupervised manifold clustering," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on.* IEEE, 2007, pp. 1–6.

[62] D. Goldfarb, S. Ma, and K. Scheinberg, "Fast alternating linearization methods for minimizing the sum of two convex functions," Math. Program. Ser. A, in Press, 2010.

[63] J. Grosek and J. N. Kutz, "Dynamic mode decomposition for real-time background/foreground separation in video," *arXiv preprint arXiv:1404.7592*, 2014.

[64] A. Gruber and Y. Weiss, "Multibody factorization with uncertainty and missing data using the em algorithm," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1. IEEE, 2004, pp. I–707.

[65] Y. Guo and W. Xue, "Probabilistic multi-label classification with sparse feature learning," in *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence.* AAAI Press, 2013, pp. 1373–1379.

[66] C. Guyon, T. Bouwmans, and E. Zahzah, "Foreground detection based on low-rank and block-sparse matrix decomposition," in *International Conference on Image Processing, ICIP 2012*, 2012.

[67] ——, *Robust Principal Component Analysis for Background Subtraction: Systematic Evaluation and Comparative Analysis*, 2012, pp. 223–238.

[68] C. Guyon, T. Bouwmans, and E.-H. Zahzah, "Foreground detection via robust low-

rank matrix decomposition including spatio-temporal constraint," in *International Workshop on Background Model Challenges, ACCV 2012,*, 2012, pp. 315–320.

[69] T. Haines and T. Xiang, "Background subtraction with dirichlet-process mixture models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 4, pp. 670–683, April 2014.

[70] R. Hartley and F. Schaffalitzky, "Powerfactorization: 3d reconstruction with missing or uncertain data," in *Australia-Japan advanced workshop on computer vision*, vol. 74, 2003, pp. 76–85.

[71] J. He, L. Balzano, and A. Szlam, "Incremental gradient on the grassmannian for online foreground and background separation in subsampled video," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on.* IEEE, 2012, pp. 1568–1575.

[72] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *arXiv preprint arXiv:1703.06870*, 2017.

[73] M. Hofmann, P. Tiefenbacher, and G. Rigoll, "Background segmentation with feedback: The pixel-based adaptive segmenter," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on.* IEEE, 2012, pp. 38–43.

[74] Y. P. Hong and C.-T. Pan, "Rank-revealing QR factorizations and the singular value decomposition," *Mathematics of Computation*, vol. 58, no. 197, pp. 213–232, 1992.

[75] D. Hsu, S. M. Kakade, and T. Zhang, "Robust matrix decomposition with sparse corruptions," *IEEE Transactions on Information Theory*, vol. 57, no. 11, pp. 7221–7234, 2011.

[76] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.

[77] J. Huang, X. Huang, and D. Metaxas, "Learning with dynamic group sparsity," in *Computer Vision, 2009 IEEE 12th International Conference on.* IEEE, 2009, pp. 64–71.

[78] H. ITU-T RECOMMENDATION, "264 advanced video coding for generic audio-visual services," 2003.

[79] S. Javed, S. Oh, A. Sobral, T. Bouwmans, and S. Jung, "Background subtraction via superpixel-based online matrix decomposition with structured foreground constraints," in *Workshop on Robust Subspace Learning and Computer Vision, ICCV 2015*, 2015.

[80] R. Jenatton, J.-Y. Audibert, and F. Bach, "Structured variable selection with sparsity-inducing norms," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2777–2824, 2011.

[81] K. Jia, T.-H. Chan, and Y. Ma, "Robust and practical face recognition via structured sparsity," in *Computer Vision–ECCV 2012*. Springer, 2012, pp. 331–344.

[82] I. T. Jolliffe, "Discarding variables in a principal component analysis. i: Artificial data," *Applied statistics*, pp. 160–173, 1972.

[83] M. Karaman, L. Goldmann, D. Yu, and T. Sikora, "Comparison of static background segmentation methods," in *Visual Communications and Image Processing 2005*. International Society for Optics and Photonics, 2005, pp. 596 069–596 069.

[84] T. Kato, H. Hino, and N. Murata, "Multi-frame image super resolution based on sparse coding," *Neural Networks*, vol. 66, pp. 64–78, 2015.

[85] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected CRFs with Gaussian edge potentials," in *Advances in neural information processing systems*, 2011, pp. 109–117.

[86] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[87] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," *Advances in neural information processing systems*, vol. 19, p. 801, 2007.

[88] K.-C. Lee, J. Ho, and D. J. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *Pattern Analysis and Machine Intelligence, IEEE*

Transactions on, vol. 27, no. 5, pp. 684–698, 2005.

[89] C.-G. Li, C. You, and R. Vidal, "Structured sparse subspace clustering: A joint affinity learning and subspace clustering framework," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2988–3001, 2017.

[90] L. Li, W. Huang, I. Y. Gu, and Q. Tian, "Foreground object detection from videos containing complex background," in *Proceedings of the eleventh ACM international conference on Multimedia*. ACM, 2003, pp. 2–10.

[91] L. Li, W. Huang, I. Y.-H. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," *IEEE Transactions on Image Processing*, vol. 13, no. 11, pp. 1459–1472, 2004.

[92] L. Li, W. Huang, I.-H. Gu, and Q. Tian, "Statistical modeling of complex backgrounds for foreground object detection," *Image Processing, IEEE Transactions on*, vol. 13, no. 11, pp. 1459–1472, Nov 2004.

[93] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, *Microsoft COCO: Common Objects in Context*. Cham: Springer International Publishing, 2014, pp. 740–755. [Online]. Available: https://doi.org/10.1007/978-3-319-10602-1_48

[94] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *arXiv preprint arXiv:1009.5055*, 2010.

[95] Z. Lin, M. Chen, L. Wu, and Y. Ma, "The augmented lagrange multiplier method for exact recovey of corrupted low-rank matrices," UIUC Technical Report, Tech. Rep., 2009.

[96] Z. Lin, A. Ganesh, J. Wright, L. Wu, M. Chen, and Y. Ma, "Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix," in *In International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, Aruba, Dutch Antilles*, 2009.

[97] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 1, pp. 171–184, 2013.

[98] J. Liu, S. Ji, and J. Ye, "Multi-task feature learning via efficient $\ell_{2,1}$-norm mini-

mization," in *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 2009, pp. 339–348.

[99] J. Liu and J. Ye, "Moreau-Yosida regularization for grouped tree structure learning," in *Advances in Neural Information Processing Systems*, 2010, pp. 1459–1467.

[100] R. Liu, Z. Lin, S. Wei, and Z. Su, "Solving principal component pursuit in linear time via ℓ_1 filtering," *CoRR*, vol. abs/1108.5359, 2011. [Online]. Available: http://arxiv.org/abs/1108.5359

[101] X. Liu, G. Zhao, J. Yao, and C. Qi, "Background subtraction based on low-rank and structured sparse decomposition," 2015.

[102] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang, "Semantic image segmentation via deep parsing network," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1377–1385.

[103] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[104] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[105] S. Ma, "Algorithms for sparse and low-rank optimization: Convergence complexity and applications thesis," Ph.D. dissertation, June 2011.

[106] Y. Ma, "Pursuit of low-dimensional structures in high-dimensional visual data," *Plenary talk at the Foundations of Computational Mathematics, FoCM 2014*, 2014.

[107] Y. Ma, J. Wright, and A. Y. Yang, "Sparse and low-dimensional representation, lecture 3: Modeling high-dimensional (Visual) data," 2012.

[108] Y. Ma, A. Y. Yang, H. Derksen, and R. Fossum, "Estimation of subspace arrangements with applications in modeling and segmenting mixed data," *SIAM review*, vol. 50, no. 3, pp. 413–458, 2008.

[109] L. Maddalena and A. Petrosino, "A self-organizing approach to background subtraction for visual surveillance applications," *IEEE Transations on Image Processing*, vol. 17, no. 7, pp. 1168–1177, July 2008. [Online]. Available: http://dx.doi.org/10.1109/TIP.2008.924285

[110] ——, "The SOBS algorithm: what are the limits?" in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. IEEE, 2012, pp. 21–26.

[111] M. W. Mahoney and P. Drineas, "CUR matrix decompositions for improved data analysis," *Proceedings of the National Academy of Sciences*, vol. 106, no. 3, pp. 697–702, 2009.

[112] J. Mairal, R. Jenatton, F. R. Bach, and G. R. Obozinski, "Network flow algorithms for structured sparsity," in *Advances in Neural Information Processing Systems*, 2010, pp. 1558–1566.

[113] G. Mateos and G. B. Giannakis, "Sparsity control for robust principal component analysis," in *Signals, Systems and Computers (ASILOMAR), 2010 Conference Record of the Forty Fourth Asilomar Conference on*. IEEE, 2010, pp. 1925–1929.

[114] ——, "Robust PCA as bilinear decomposition with outlier-sparsity regularization," *IEEE Transactions on Signal Processing*, vol. 60, no. 10, pp. 5176–5190, 2012.

[115] Y. Mu, J. Dong, X. Yuan, and S. Yan, "Accelerated low-rank visual recovery by random projection," in *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 2609–2616. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2011.5995369

[116] S. Nakajima, M. Sugiyama, and S. D. Babacan, "Sparse additive matrix factorization for robust PCA and its generalization," in *Asian Conference on Machine Learning*, 2012, pp. 301–316.

[117] F. Nie, H. Huang, X. Cai, and C. H. Ding, "Efficient and robust feature selection via joint 2, 1-norms minimization," in *Advances in neural information processing systems*, 2010, pp. 1813–1821.

[118] G. Obozinski, B. Taskar, and M. Jordan, "Multi-task feature selection," *Statistics Department, UC Berkeley, Tech. Rep*, 2006.

[119] J.-M. Odobez and P. Bouthemy, "Robust multiresolution estimation of parametric motion models," *Journal of visual communication and image representation*, vol. 6, no. 4, pp. 348–365, 1995.

[120] N. M. Oliver, B. Rosario, and A. P. Pentland, "A Bayesian computer vision system for modeling human interactions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 831–843, 2000.

[121] O. Oreifej, "Robust subspace estimation using low-rank optimization. theory and applications in scene reconstruction, video denoising, and activity recognition." 2013.

[122] O. Oreifej and M. Shah, *Robust Subspace Estimation Using Low-Rank Optimization.* Springer, 2014.

[123] D. Papailiopoulos, A. Kyrillidis, and C. Boutsidis, "Provable deterministic leverage score sampling," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining.* ACM, 2014, pp. 997–1006.

[124] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma, "RASL: robust alignment by sparse and low-rank decomposition for linearly correlated images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2233–2246, 2012.

[125] C. Qiu and N. Vaswani, "ReProCS: A missing link between recursive robust PCA and recursive sparse recovery in large but correlated noise," *arXiv preprint arXiv:1106.3286*, 2011.

[126] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam, "Image change detection algorithms: A systematic survey," *IEEE Transactions on Image Processing*, vol. 14, pp. 294–307, 2005.

[127] S. Rao, R. Tron, R. Vidal, and Y. Ma, "Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 10, pp. 1832–1845, 2010.

[128] S. R. Rao, A. Y. Yang, S. S. Sastry, and Y. Ma, "Robust algebraic segmentation of mixed rigid-body and planar motions from two views," *International journal of computer vision*, vol. 88, no. 3, pp. 425–446, 2010.

[129] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[130] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.

[131] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[132] H. Sajid and S.-C. S. Cheung, "Background subtraction for static & moving camera," in *Image Processing (ICIP), 2015 IEEE International Conference on*.

[133] ——, "Universal multimode background subtraction," in *Image Processing (ICIP), Submitted to 2015 IEEE International Conference on*.

[134] A. Schick, M. Bäuml, and R. Stiefelhagen, "Improving foreground segmentations with probabilistic superpixel markov random fields," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on*. IEEE, 2012, pp. 27–31.

[135] B. Schölkopf, A. Smola, and K.-R. Müller, "Kernel principal component analysis," in *International Conference on Artificial Neural Networks*. Springer, 1997, pp. 583–588.

[136] Y. Shen, Z. Wen, and Y. Zhang, "Augmented lagrangian alternating direction method for matrix separation based on low-rank factorization," *Optimization Methods Software*, vol. 29, no. 2, pp. 239–263, Mar. 2014. [Online]. Available: http://dx.doi.org/10.1080/10556788.2012.700713

[137] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1874–1883.

[138] W. Siming and L. Zhouchen, "Analysis and improvement of low-rank representation for subspace segmentation," *arXiv preprint arXiv:1107.1561*, 2011.

[139] L. Song, X. Tang, W. Zhang, X. Yang, and P. Xia, "The SJTU 4K video sequence dataset," in *Quality of Multimedia Experience (QoMEX), 2013 Fifth International*

Workshop on. IEEE, 2013, pp. 34–35.

[140] P.-L. St-Charles, G.-A. Bilodeau, and R. Bergevin, "A self-adjusting approach to change detection based on background word consensus," in *Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on*. IEEE, 2015, pp. 990–997.

[141] ——, "SuBSENSE: A universal change detection method with local adaptive sensitivity," *Image Processing, IEEE Transactions on*, vol. 24, no. 1, pp. 359–373, 2015.

[142] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2. IEEE, 1999.

[143] M. Stojnic, F. Parvaresh, and B. Hassibi, "On the reconstruction of block-sparse signals with an optimal number of measurements," *Signal Processing, IEEE Transactions on*, vol. 57, no. 8, pp. 3075–3085, 2009.

[144] Y. Sugaya and K. Kanatani, "Multi-stage unsupervised learning for multi-body motion segmentation," *IEICE Transactions on Information and Systems*, vol. 87, no. 7, pp. 1935–1942, 2004.

[145] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[146] V. Sze, M. Budagavi, and G. J. Sullivan, "High efficiency video coding (HEVC)," *Integrated Circuit and Systems, Algorithms and Architectures*, pp. 1–375, 2014.

[147] R. Szeliski, *Computer Vision: Algorithms and Applications*, 1st ed. New York, NY, USA: Springer-Verlag New York, Inc., 2010.

[148] G. Tang and A. Nehorai, "Robust principal component analysis based on low-rank and block-sparse matrix decomposition," in *45th Annual Conference on Information Sciences and Systems, CISS, The John Hopkins University, Baltimore, MD, USA, 23-25 March 2011*, 2011, pp. 1–5.

[149] M. Tao and X. Yuan, "Recovering low-rank and sparse components of matrices from incomplete and noisy observations," *SIAM Journal on Optimization*, vol. 21,

no. 1, pp. 57–81, 2011.

[150] M. E. Tipping and C. M. Bishop, "Probabilistic principal component analysis," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 61, no. 3, pp. 611–622, 1999.

[151] F. D. L. Torre and M. J. Black, "A framework for robust subspace learning," *International Journal of Computer Vision*, vol. 54, p. 2003, 2003.

[152] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers, "Wallflower: principles and practice of background maintenance," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 1, 1999, pp. 255–261 vol.1.

[153] R. Tron and R. Vidal, "A benchmark for the comparison of 3-D motion segmentation algorithms," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.

[154] A. Vedaldi, G. Guidi, and S. Soatto, "Joint data alignment up to (lossy) transformations," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[155] R. Vidal and R. Hartley, "Motion segmentation with missing data using PowerFactorization and GPCA," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 2. IEEE, 2004, pp. II–310.

[156] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (GPCA)," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1945–1959, 2005.

[157] R. Vidal, R. Tron, and R. Hartley, "Multiframe motion segmentation with missing data using PowerFactorization and GPCA," *International Journal of Computer Vision*, vol. 79, no. 1, pp. 85–105, 2008.

[158] A. Wagner, J. Wright, A. Ganesh, Z. Zhou, H. Mobahi, and Y. Ma, "Toward a practical face recognition system: Robust alignment and illumination by sparse representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 2, pp. 372–386, 2012.

[159] N. Wang and D.-Y. Yeung, "Bayesian robust matrix factorization for image and

video processing," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, Dec 2013, pp. 1785–1792.

[160] Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, "CDnet 2014: An Expanded Change Detection Benchmark Dataset," in *IEEE CVPR Change Detection workshop*, United States, Jun. 2014, p. 8 p., https://hal-univ-bourgogne.archives-ouvertes.fr/hal-01018757.

[161] R. Weerakkody and M. Mrak, "High efficiency video coding for ultra high definition television," *Proc. 2013 NEM Summit*, pp. 9–14, 2013.

[162] B. Wohlberg, R. Chartrand, and J. Theiler, "Local principal component pursuit for non-linear datasets," in *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Kyoto, Japan, Mar. 2012, pp. 3925–3928.

[163] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009. Proceedings of a meeting held 7-10 December 2009, Vancouver, British Columbia, Canada.*, 2009, pp. 2080–2088.

[164] S. Wu, O. Oreifej, and M. Shah, "Action recognition in videos acquired by a moving camera using motion decomposition of lagrangian particle trajectories," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1419–1426.

[165] B. Xin, Y. Tian, Y. Wang, and W. Gao, "Background subtraction via generalized fused Lasso foreground modeling," *arXiv preprint arXiv:1504.03707*, 2015.

[166] H. Xu, C. Caramanis, and S. Sanghavi, "Robust PCA via outlier pursuit," *IEEE Transactions on Information Theory*, vol. 58, no. 5, pp. 3047–3064, 2012.

[167] J. Yan and M. Pollefeys, "A general framework for motion segmentation: Independent, articulated, rigid, non-rigid, degenerate and non-degenerate," in *Computer Vision–ECCV 2006*. Springer, 2006, pp. 94–106.

[168] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE transactions on image processing*, vol. 19, no. 11, pp. 2861–

2873, 2010.

[169] J. Yang and X. Yuan, "Linearized augmented lagrangian and alternating direction methods for nuclear norm minimization," *Mathematics of Computation*, 2010.

[170] Y. Yang, H. T. Shen, Z. Ma, Z. Huang, and X. Zhou, "$\ell_{2,1}$-norm regularized discriminative feature selection for unsupervised learning," in *IJCAI Proceedings-International Joint Conference on Artificial Intelligence*, vol. 22, no. 1. Citeseer, 2011, p. 1589.

[171] X. Yuan and J. Yang, "Sparse and low-rank matrix decomposition via alternating direction methods," Tech. Rep., 2009.

[172] L. Zelnik-Manor, K. Rosenblum, and Y. C. Eldar, "Dictionary optimization for block-sparse representations," *Signal Processing, IEEE Transactions on*, vol. 60, no. 5, pp. 2386–2395, 2012.

[173] T. Zhang, A. Szlam, Y. Wang, and G. Lerman, "Hybrid linear modeling via local best-fit flats," *International journal of computer vision*, vol. 100, no. 3, pp. 217–240, 2012.

[174] Y. Zhang, "An alternating direction algorithm for non-negative matrix factorization," *preprint*, 2010.

[175] Z. Zhang and V. Sze, "FAST: Free adaptive super-resolution via transfer for compressed videos," *arXiv preprint arXiv:1603.08968*, 2016.

[176] T. Zhou and D. Tao, "GoDec: Randomized low-rank and sparse matrix decomposition in noisy case," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, ser. ICML '11, L. Getoor and T. Scheffer, Eds. ACM, June 2011, pp. 33–40.

[177] ——, "Shifted subspaces tracking on sparse outlier for motion segmentation," in *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9*, 2013.

[178] X. Zhou, C. Yang, and W. Yu, "DECOLOR: Moving object detection by detecting contiguous outliers in the low-rank representation." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 3, pp. 597–610, 2013.

[179] Z. Zhou, X. Li, J. Wright, E. J. Candès, and Y. Ma, "Stable principal

component pursuit," *CoRR*, vol. abs/1001.2363, 2010. [Online]. Available: http://arxiv.org/abs/1001.2363

[180] Z. Zivkovic, "Improved adaptive gaussian mixture model for background subtraction," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 2. IEEE, 2004, pp. 28–31.

[181] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern recognition letters*, vol. 27, no. 7, pp. 773–780, 2006.